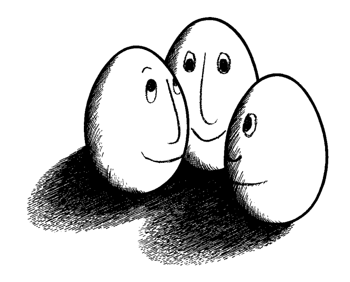
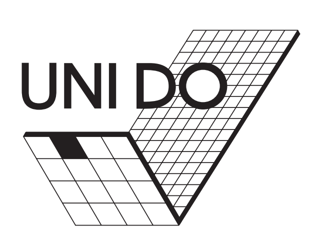
Entwicklung eines wissensbasierten Assistentensystems zur Analyse von Fall–Kontroll–Studien

Ursula Robers

betreut von
Prof. Dr. Katharina Morik
und
Prof. Dr. Iris Pigeot–Kübler

Juli 1995





Diplomarbeit am Fachbereich Informatik der Universität Dortmund

Vorwort

An erster Stelle möchte ich meinen beiden Betreuerinnen Prof. Dr. Katharina Morik und Prof. Dr. Iris Pigeot–Kübler für ihr großes Interesse und das Engagement bei der Betreuung dieser Arbeit danken. Insbesondere danke ich Frau Pigeot–Kübler als Leiterin des Projekts CORA, in dessen Rahmen diese Arbeit entstanden ist, für die hervorragende Zusammenarbeit und das in mich gesetzte Vertrauen.
Desweiteren gilt mein besonderer Dank Ursula Sondhauß, der wissenschaftlichen Mitarbeiterin im CORA–Projekt, die durch eine Reihe von wertvollen Vorschlägen und Ideen sowie durch stete Geprächsbereitschaft diese Arbeit in besonderem Maße gefördert hat.
Auch den anderen Mitarbeitern des Projekts sowie den Teilnehmern des Diplomandenseminars sei herzlich gedankt.

Essen, im Juli 1995 Ursula Robers

Zusammenfassung

Das Thema der vorliegenden Diplomarbeit ist die Entwicklung eines Systems, das eine statistische Analyse bestimmter epidemiologischer Studien, sog. Fall–Kontroll–Studien, ermöglicht. Da für diese Analyse umfangreiche Kenntnisse statistischer Verfahren und ihrer Eigenschaften erforderlich ist, diese aber in der Regel von Epidemiologen durchgeführt wird, lag die Idee nahe, das System mit zusätzlichem „Wissen“ auszustatten, um seinen Benutzer bei der Analyse umfassend zu unterstützen. Für die Entwicklung waren daher drei Aspekte wesentlich. Das System sollte mit einem benutzerfreundlichen Interface ausgestattet sein. Desweiteren wurde ein Hilfesystem konzipiert, das jeweils allgemeine und auch kontext–sensitive Hilfen sowohl zu programm–technischen Fragen als auch zu Fragen inhaltlicher Art anbietet. Als dritter Punkt wurde eine Beratung des Benutzers im Sinne einer Entscheidungsunterstützung in Form einer Beratungskomponente umgesetzt. Diese Komponente des Systems beruht auf einer wissensbasierten Auswertung von simulierten Fall–Kontroll–Studien, sog. Simulationsstudien oder auch Monte–Carlo–Studien. Aus diesen Studien wurde mithilfe des maschinellen Lernens eine Charakterisierung von einigen im Rahmen der Kontingenztafelanalyse zu benutzenden statistischen Methoden gewonnen. Diese Charakterisierung in Form einer Regelmenge schließt von bestimmten Dateneigenschaften einer Fall–Kontroll–Studie auf die Eignung der unterschiedlichen statistischen Methoden. Die Beratungskomponente des Systems besteht damit aus einer Regelmenge, die in einer gegebenen Datensituation eine geeignete Methode zur Datenanalyse vorschlägt.

Abstract

This work describes the development of a system for a statistical analysis of case control studies. The analysis of epidemiological studies requires high level statistical knowledge, such that the users of the system, mainly epidemiologists, should be assisted. This is realized by guiding them through all phases of the analysis and by providing them with background knowledge. Therefore three main aspects have to be part of the development of the system. First the grafical user/system interface should be user–friendly. Second a wide range help system has to be developed that gives the user not only information about how to use the program but also an introduction to the underlying statistical background, i.e. the contingency table analysis. The third aspect is to build an advice component that assists the user in choosing an appropriate statistical method for their analysis.

This work focuses on the process of getting the knowledge that is required to build such an advice component. Here, we explore simulation studies[1] to gain information about the different statistical methods typically applied for analyzing the data. With the help of machine learning we get a characterization of the examined statistical methods in form of a ruleset. The rules infer the properties of a certain statistical method from the data attributes of the case control study. The advice component can now assist the user by providing recommendations according to these rules which can be regarded as guidelines.

[1] simulated case control study

Inhaltsverzeichnis

Kapitel 1

Einleitung

Im letzten Jahrzehnt ist eine ständig wachsende Bedeutung statistischer Methoden für viele Bereiche der Wissenschaft, insbesondere der Medizin und Epidemiologie, zu beobachten. Einerseits führten die Fortschritte auf dem Gebiet der Computertechnologie durch die Konstruktion immer leistungsfähigerer Rechner zu einer verstärkten Entwicklung komplexer statistischer Verfahren. Andererseits erfordert die steigende Komplexität des Datenmaterials die Anwendung rechenintensiver statistischer Methoden, deren Einsatz überhaupt erst durch den Computer möglich wird.
Es wurde bereits früh erkannt, daß sich aus der Verwendung rechnergestützter Systeme neue Fehlermöglichkeiten ergeben: durch den Einsatz hochentwickelter statistischer Auswertungssysteme wie beispielsweise SAS wird die Verwendung von komplexen Verfahren immer einfacher. Jedoch führt die Benutzung dieser Systeme durch Wissenschaftler ohne Expertenwissen aus der Statistik häufig zu Fehlentscheidungen. Die Fehler betreffen hier neben der Auswahl der Verfahren auch die Interpretation der Ergebnisse. Außerdem wird durch mangelnde Vertrautheit mit den zur Verfügung stehenden Methoden häufig auf „Altbewährtes“ zurückgegriffen, obwohl der Einsatz eines anderen Verfahrens in der gegebenen Situation vielleicht günstiger wäre. Selbst bei dem Einsatz adäquater Methoden können bei Nichtbeachtung des statistischen Hintergrunds Fehler entstehen. In [Gebhardt, 1988] wird auf folgendes Problem hingewiesen:

> „Da es für einen gegebenen Datensatz nicht nur eine einzige Möglichkeit der statistischen Auswertung gibt, besteht die Gefahr, daß der Benutzer mehrere Analysen durchführt und sich dann die „passenden“ Ergebnisse heraussucht und die anderen ignoriert“.

Daraus ergibt sich die Forderung nach Systemen, die mittels zusätzlichem Wissen eine „sinnvolle“ Anwendung statistischer Methoden unterstützen. Zudem müssen die Auswertungssys-

teme die Interpretation der Ergebnisse durch abrufbare Hilfestellungen für den Anwender verdeutlichen.

Für die Konstruktion von Systemen, die diesen Anforderungen entsprechen, wurde bereits häufig der Einsatz von Methoden aus der Künstlichen Intelligenz gefordert. Insbesondere durch Verfahren, die den Modellierungsprozeß für Expertenwissen unterstützen, kann das notwendige Wissen für solche Systeme aufbereitet und verwendbar gemacht werden. Es können damit wissensbasierte Systeme entwickelt werden, die sich von der ersten Expertensystem–Generation vor allem durch die Wissensmodellierung unterscheiden. Den bereits ab Mitte der achtziger Jahre konstruierten Expertensystemen für statistische Probleme (siehe Kapitel 6) liegt eine damals weit verbreitete Transfersichtweise der Wissensakquisition zugrunde, die davon ausgeht, daß das Expertenwissen mithilfe geeigneter Tools (vorwiegend Expertensystemshells) problemlos vom Experten in das System übertragen werden kann. Daß sich diese Systeme nicht durchsetzen konnten — kaum ein solches XPS wird in der Praxis in größerem Umfang verwendet —, mag wohl auch an dieser Fehleinschätzung liegen.

Der im Rahmen dieser Arbeit vorgestellte Ansatz versucht, dieses Problem durch eine Betonung der Modellierungsphase des Wissens zu lösen. Der Modellierungsprozeß wird durch ein System unterstützt, das den manuellen Wissenserwerb mit dem maschinellen Lernen verbindet. Diese Kombination ermöglicht, daß zusätzliches Wissen, also Wissen, das nicht bereits in Form eines Modells vorliegt, aus Beobachtungen abgeleitet werden kann. Das vorliegende System CORA[1] ist mit einer Komponente zur Beratung von Benutzern bezüglich der Methodenauswahl ausgestattet, die auf einer gelernten Menge von Regeln basiert. Dazu wird eine Wissensbasis aufgebaut, die eigens zum Erwerb von Kenntnissen über Eigenschaften statistischer Methoden durchgeführte Simulationsstudien repräsentiert. In diesen Studien kann also der Zusammenhang zwischen den Eigenschaften der Daten und dem Verhalten der statistischen Methoden entdeckt bzw. gelernt werden. Die Beratungskomponente überprüft die vorliegende Datenlage und wendet dann gegebenenfalls eine passende Regel an, die eine geeignete Methode vorschlägt.

Das System beschränkt sich auf einen kleinen, aber dennoch für einen Wissenschaftler mit relativ geringen Statistikkenntnissen schwer handhabbaren Bereich der Statistik, nämlich auf die Analyse geschichteter 2x2–Kontingenztafeln, mit der eine Auswertung von bestimmten epidemiologischen Studien, sogenannten Fall–Kontroll–Studien, durchgeführt werden kann. Mithilfe dieser Studien können Risikofaktoren für Gesundheitsschäden großer Bevölkerungsgruppen qualitativ sowie quantitativ erfaßt werden.

Im Rahmen dieser Analyse sind eine Reihe von Auswahlentscheidungen durch den Benutzer zu treffen, wobei die Methodenauswahl für die Abschätzung des Risikos einer Erkrankung von

[1] Combined Odds Ratio Analysis

zentraler Bedeutung ist. Diese Entscheidung kann durch die oben erwähnte Beratungskomponente vom System unterstützt werden, die, wie bereits erwähnt, auf der wissensbasierten Auswertung simulierter Fall–Kontroll–Studien aufbaut.

Ein weiterer wichtiger Aspekt des Systems sind die angebotenen Hilfestellungen für den Benutzer. Hier werden sowohl die statistischen Hintergünde, die u.a. zur Interpretation der Ergebnisse erforderlich sind, erläutert als auch teilweise kontext–sensitive Hilfen zur Programmbenutzung gegeben.

Das gesamte System ist mit einer einheitlichen grafischen Oberfläche ausgestattet, auf deren Benutzerfreundlichkeit besonderer Wert gelegt wird. Alle Komponenten des Systems können von dieser Oberfläche aufgerufen werden. Dazu gehören auch die statistischen Verfahren, die für die beschriebene Analyse der Daten erforderlich sind. Die zugehörigen Prozeduren sind direkter Bestandteil des Systems, d.h. es handelt sich hier nicht um eine „intelligente Schnittstelle“ (siehe auch Kapitel 6) zu einem vorhandenen statistischen Softwarepaket, sondern um ein völlig eigenständiges System.

Die Konzeption und Entwicklung dieses wissensbasierten Assistentensystems CORA wird in der vorliegenden Arbeit spezifiziert und ausführlich beschrieben.

Die Arbeit umfaßt insgesamt sieben Kapitel. Im Anschluß an diese Einleitung werden im Kapitel 2 zunächst die verwendeten Hilfsmittel zur Konstruktion des wissensbasierten Systems (WBS) vorgestellt. Es handelt sich dabei um KADS, eine Methodik, die den Knowledge Engineering Prozeß unterstützt. Mithilfe von KADS wird das WBS spezifiziert. Desweiteren wird das System MOBAL vorgestellt, das den Wissenserwerbsprozeß durch maschinelles Lernen unterstützt und ein operationales Modell des Wissens zur Verfügung stellt. Mit dem System Delphi wird eine Entwicklungsumgebung zur objekt–orientierten Konstruktion von Anwendungen eingesetzt, die eine umfangreiche Komponentenbibliothek zur Verfügung stellt.

Der KADS–Ansatz unterscheidet für die Spezifikation des WBS eine Reihe von Modellen, die auf unterschiedlichen Ebenen liegen. Diese werden in Kapitel 2.1 eingeführt. Zunächst werden die Modelle auf dem Knowledge Level, also der implementationsunabhängigen Ebene, beschrieben. Hier wird das vom System zu leistende Problemlösungsverhalten detailliert vorgestellt. Das Organisations–, das Anwendungs–, das Expertisen– und das Kooperationsmodell beleuchten die unterschiedlichen Aspekte des Sachbereichs.

Kapitel 4 geht dann auf die durch das maschinelle Lernen unterstützte Modellierung der Expertise genauer ein. Dabei wird der Modellierungs– und Lernzyklus beschrieben, an dessen Ende eine, vorwiegend gelernte, Regelmenge steht, die in das WBS eingebunden wird.

Kapitel 5 spezifiziert im Anschluß das Designmodell für das System CORA, also ein Symbol Level Modell, das die Architektur und das Design der einzelnen Systemkomponenten vorstellt. Dabei werden hauptsächlich die konzeptuellen Entscheidungen beschrieben und nur an

einzelnen Stellen konkrete Implementationsdetails zur Verdeutlichung angegeben.
Anschließend wird das System CORA bewertet und mit anderen Ansätzen verglichen. Kapitel 7 faßt die wesentlichen Ergebnisse noch einmal kurz zusammen und zeigt daraufhin mögliche Perspektiven sowohl für die Weiterentwicklung dieses Systems als auch für intelligentere statistische Auswertungssoftware insgesamt auf.

Kapitel 2

Hilfsmittel zur Konstruktion des Systems CORA: KADS, MOBAL und Delphi

Im folgenden Kapitel werden die Systeme, die zur Konstruktion des wissensbasierten Assistentensystems CORA verwendet werden, vorgestellt. Es handelt sich hierbei zunächst um KADS, eine Methodik für den Entwurf wissensbasierter Systeme. Mithilfe dieser Methodik wird eine Spezifikation des Systems CORA entworfen. Die Modellierung des Expertenwissens und das Lernen einer Charakterisierung von Schätzern ist mit Unterstützung des Modellierungssystems MOBAL durchgeführt worden. Für die Implementierung von CORA wurde ein objektorientierter Ansatz gewählt, der auf das Tool Delphi zurückgreift. Diese drei Systeme werden im folgenden vorgestellt.

2.1 KADS

KADS[1] ist eine strukturierte Methodik für die Konstruktion wissensbasierter Systeme (WBS), die ausführlich in [Schreiber et al, 1993] beschrieben wird. Das Knowledge Engineering wurde lange Zeit vom Rapid Prototyping, also dem Bau von Prototypen (siehe beispielsweise [Karbach, 1989]), bestimmt. Der KADS–Ansatz wendet sich entschieden von dieser Methodik ab. Im Mittelpunkt des Knowledge Engineerings und damit der Konstruktion wissensbasierter Systeme steht im Rahmen von KADS der Knowledge Akquisition Prozeß. Die Erkenntnis, daß eine *Modellierung* des Expertenwissens stattfinden muß, ist von zentraler Bedeutung und widerspricht der Transfersichtweise, die dem Rapid Prototyping zugrunde liegt. Der KADS–

[1] Knowledge Acquisition and Documentation System

Ansatz baut auf zwei grundlegenden Prinzipien auf:

- der Einführung von Zwischenmodellen und
- der Einführung eines Knowledge Level Modells, das vom Symbol Level getrennt wird.

Im folgenden Abschnitt werden zunächst die Sichtweisen der Wissensakquisition, nämlich die Transfersichtweise, die Modellierungssichtweise des KADS–Ansatzes und die Modellierungssichtweise des Sloppy Modeling–Ansatzes (siehe [Morik, 1989]) vorgestellt. In dem anschließenden Abschnitt werden die beiden grundlegenden Prinzipien der KADS–Methodologie erläutert. In 2.1.3 findet sich ein kurzer Überblick über die einzelnen Modelle. Das wichtigste Modell, das Expertisenmodell, wird in 2.1.4 detailliert beschrieben. Danach folgt ein Abschnitt über die Modellierung der Kooperation zwischen dem Benutzer und dem System, die im Rahmen der Entwicklung wissensbasierter Systeme einen besonderen Stellenwert besitzt. Eine Diskussion des KADS–Ansatzes schließt dieses Kapitel ab.

2.1.1 Sichtweisen der Knowledge Acquisition

Als das zentrale Problem der Konstruktion wissensbasierter Systeme wird häufig das sogenannte Knowledge Acquisition Bottleneck angesehen, das darin besteht, das Wissen aus dem Experten „herauszuholen“ und als komplettes Modell in das wissensbasierte System „hineinzustecken“. Hier wird deutlich, daß die Wissenakquisition auf den Transfer des Wissens vom Experten zum System reduziert wird.
Neuere Ansätze gehen dazu über, die Knowledge Acquisition als einen Modellierungsprozeß zu betrachten. Diese Ansätze setzen nicht das Vorhandensein eines fertigen Modells der Expertise voraus. Die Wissensakquisition wird vielmehr als Modellbildung verstanden. Der zweite Modellierungsansatz, der hier vorgestellt wird, betont die Möglichkeit, das Modell zu revidieren und verbindet damit Ideen des Rapid Prototyping Ansatzes mit der Modellierungssichtweise der Knowledge Acquisition.

Die Transfersicht der Knowledge Acquisition

Im Rahmen der Transfersichtweise wird davon ausgegangen, daß ein fertiges Modell der Expertise vorliegt, also die Task Performance des Experten auf einem Modell beruht. Dieses Modell muß dann nur noch aus dem Experten herausgeholt und und mit geeigneten Methoden adäquat in das wissensbasierte System übertragen werden. In der Praxis bedeutet das, daß der Experte nach Regeln gefragt wird, die in einer bestimmten Situation angewendet werden können. Der Wissensingenieur übersetzt dann die natürlichsprachlichen Regeln in ein formales Konzept.

Die Metapher des Flaschenhalses kann (laut [Morik, 1989]) auf zwei Weisen interpretiert werden:

- Das Expertensystem wird als Flasche angesehen, in die das Wissen hineingebracht werden muß, oder
- der Experte wird mit einer Flasche verglichen, wobei das Problem darin besteht, das Wissen aus ihm herauszuholen.

In älteren Expertensystemen (wie z.B. in MYCIN, siehe [Buchanan et al, 1984]) sind alle Wissensarten (Sachbereichswissen, Aufgabenwissen etc.) in der gleichen Datenstruktur repräsentiert worden, nämlich als Produktionsregeln. Zur Unterstützung der Knowledge Acquisition sind Tools entwickelt worden, die die Funktion eines Trichters übernehmen, um das Expertensystem mit Wissen zu füllen. Diese Tools können aber Mängel, die durch den Repräsentationsformalismus des Expertensystems entstehen, nicht kompensieren.

Neuere Entwicklungen konzentrierten sich darauf, Expertensystemshells mit Repräsentationskonstrukten zu entwickeln, die den Konzepten des Experten entsprechen. Ein Scheitern des Transfers wird dann auf den falschen Repräsentationsformalismus zurückgeführt. Systeme wie OPAL (siehe [Musen, 1989]) ermöglichen das direkte Eingeben des Expertenwissens in das System. Diesem System liegt aber bereits eine abgeschlossene Modellbildung zugrunde, d.h. es existiert ein fertiges Modell des Sachbereichs, hier die Krebstherapieplanerstellung, bevor die Wissensakquisition gestartet wird.

Beiden Ansätzen (MYCIN und OPAL) ist gemeinsam, daß sie das Sachbereichsmodell als gegeben ansehen. Das Problem besteht nur darin, dieses Modell von einer Repräsentation in eine andere, also in die des Systems, zu überführen. Die Transfersicht der Knowledge Acquisition ist in Abbildung 2.1 dargestellt. Die Abbildung macht deutlich, daß es keine Rückkopplung vom System bzw. der Wissensbasis zum Modell und zum Sachbereich gibt.

Wird der Experte mit der Flasche verglichen, steht der Wissenserwerb im Sinne der Knowledge Elicitation im Vordergrund. Aber auch die Knowledge Elicitation geht davon aus, daß das Modell schon fertig im Kopf des Experten vorliegt. Der Wissenserwerb wird dann als Suche nach diesem Modell verstanden, die mithilfe einer angemessenen Befragung des Experten durchgeführt werden kann.

Die Modellierungssicht der Knowledge Acquisition gemäß dem KADS–Ansatz

Die Entwicklung eines WBS wird im Rahmen des KADS–Ansatzes als ein Modellierungsprozeß angesehen. Es entsteht damit eine Modellierungssichtweise der Wissensakquisition, in der das WBS als operationales Modell betrachtet wird. Bezüglich der relevanten Eigenschaften

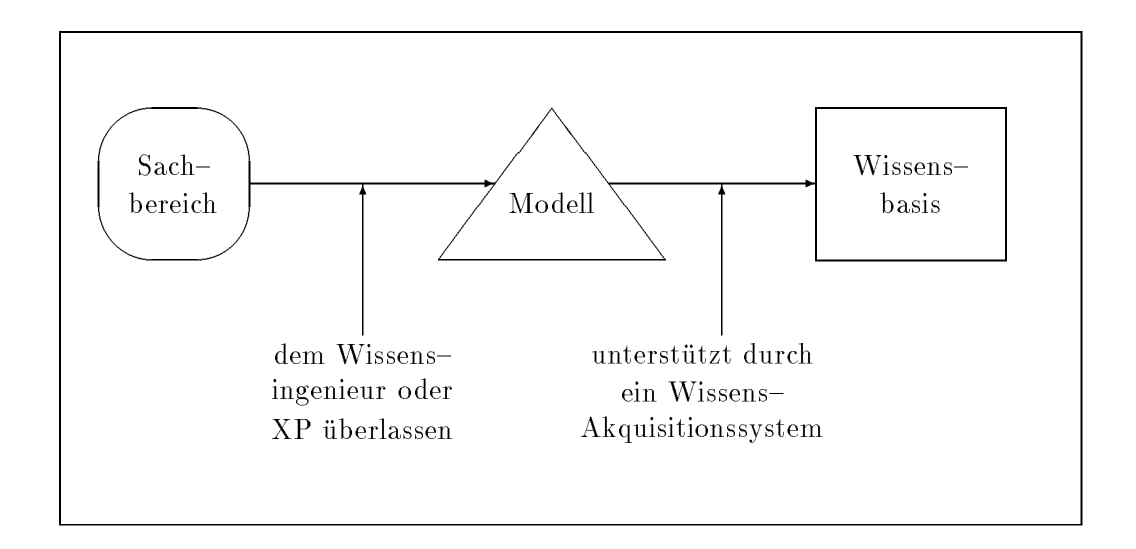


Abbildung 2.1: Die Transfersicht der Wissensakquisition nach [Morik, 1989]

sollte das Modell mit dem repräsentierten Teil der Welt übereinstimmen, also das gleiche Verhalten aufweisen. Der KADS–Ansatz wendet sich damit von der traditionellen Transfersichtweise ab.

Die Wissensakquisition des KADS–Ansatzes umfaßt im wesentlichen drei Phasen, nämlich die Wissenserhebung, nach der die Daten in einer informellen Form vorliegen, die Interpretation der Daten innerhalb eines konzeptionellen Rahmens sowie die Umsetzung dieser Konzeptualisierung auf einen symbolischen Level.

In der Modellierungssicht wird das wissensbasierte System als operationales Modell gesehen, das das gewünschte Verhalten aufweist, d.h. ein Verhalten, das dem des Experten entspricht. System und Experte können aber nicht in allen Punkten äquivalent sein: das System hat Fähigkeiten, die dem Experten fehlen, beispielsweise das Speichern sehr großer Datenmengen. Andererseits kann der Gedankengang (Reasonning Process) des Experten nicht vollständig explizit gemacht und für die Problemlösung durch das System verwendet werden.

Der Experte liefert in dieser Sichtweise lediglich die Zwänge für die Funktionalität des Systems. Der Knowledge Engineer muß in einem konstruktiven Prozeß Modellierungsentscheidungen treffen, um die einzelnen Verhaltensweisen zu einem einheitlichen System zusammenzufügen.

Gründe, warum die Modellierung im Rahmen der Wissensakquisition erforderlich ist, liefert [Morik, 1989]. Es wird festgestellt, daß die Expertise nicht in Form eines Modells vorliegen muß. Oft besteht die Kompetenz eines Experten nicht aus Wissen, sondern man kann vielmehr davon ausgehen, daß die Task Performance auf unbewußten und nicht erklärbaren Fähigkeiten (Skills) beruht. Die Techniken zur Knowledge Elicitation können daher nur erfolgreich ange-

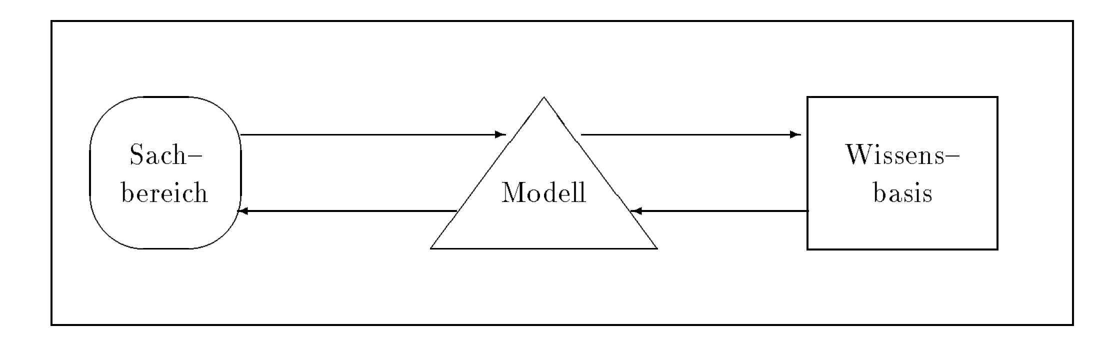


Abbildung 2.2: Wissensakquisition als Modellierung nach [Morik, 1989]

wendet werden, wenn ein Theoretiker im Gegensatz zum Task Performer interviewt wird. Fähigkeiten sollten dagegen eher beobachtet[2] als erfragt werden, da die Gefahr besteht, daß der Experte eine naive Theorie entwickelt, um seine Fähigkeiten zu erklären. Eine naive Theorie unterscheidet sich von einer wissenschaftlichen dadurch, daß sie nicht durch Nachfragen, durch das Äußern von Zweifeln und die Konfrontation mit Gegenbeispielen getestet und so verbessert wurde.

Die Aufgabe der Wissensakquisition liegt darin, die Fähigkeiten des Experten zu beschreiben. Dieser Prozeß des Beschreibens ist die Modellierung. Das Modell muß eine explizite, erklärbare und operationale Theorie des Sachbereichs darstellen.

Wie bereits erwähnt, beruft sich KADS auf diese Modellierungssichtweise der Knowledge Acquisition. Im Sinne von [Morik, 1989] liegt dem KADS–Ansatz allerdings immer noch eine Transfersichtweise zugrunde, da auch hier keine Rückkopplung vom WBS zum Modell bzw. zum Sachbereich besteht. Ein wichtiger Aspekt der Modellierungssichtweise laut [Morik, 1989], nämlich die interaktive Natur der Modellierung, wird so im KADS–Ansatz kaum berücksichtigt.

Die Modellierungssichtweise nach dem Sloppy Modeling–Prinzip

Im Sinne von Sloppy Modeling kann die Knowledge Acquisition als Modellierung wie in Abbildung 2.2 dargetellt werden. Die Modellierung ist hier ein zyklischer, kein linearer Prozeß. Der Zyklus kann in drei Phasen unterteilt werden. In der ersten Phase wird der Rahmen (Framework) für das Modell festgelegt. Es wird bestimmt, welche Aspekte des Sachbereichs relevant sind, und die zugrundeliegende Syntax und Semantik wird spezifiziert. Damit sind das Vokabular der Beschreibung von Phänomenen und semantische Beziehungen zwischen Konzepten, Eigenschaften und Zuständen bestimmt. In der nächsten Phase wird dieser Rahmen

[2] und damit als Beispiele für die Task Performance aufgefaßt werden

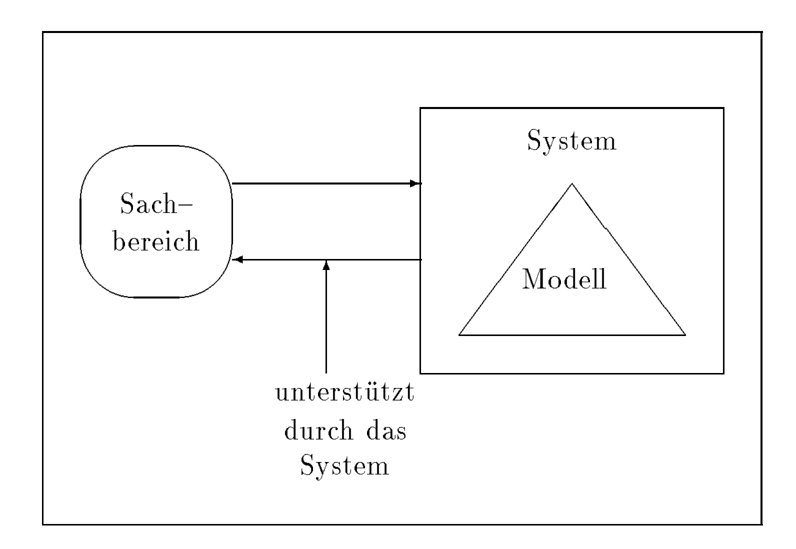


Abbildung 2.3: Operationales Modell als Ergebnis der Wissensakquisition nach [Morik, 1989]

mit Fakten und Regeln gefüllt, d.h. es werden weitere Beobachtungen repräsentiert. Können Beobachtungen nicht dargestellt werden, so muß der Rahmen revidiert werden. Die dritte Phase widmet sich der Evaluierung des Modells, indem die Konsistenz und Vollständigkeit des Modells überprüft wird. Dabei kann die Revision einzelner Fakten und Regeln, aber auch die Revision grundlegender Annahmen notwendig werden.

Ein System, das den Modellierungszyklus unterstützt, sollte dem Benutzer bei der Durchführung dieser Revisionen assistieren. Nach dem Sloppy Modeling–Prinzip ist daher keine Trennung des Knowledge Level Modells und des im System repräsentierten Modells notwendig. Die Zwischenrepräsentationen sollten in das System integriert sein (siehe Abbildung 2.3) und so eine sofortige Evaluation ermöglichen. Dieser Aspekt des Sloppy Modeling–Ansatzes entspricht der grundlegenden Idee des Rapid Prototyping. Ein System, das diesen Ansprüchen gerecht wird, ist das Modellierungssystem MOBAL (siehe 2.2).

2.1.2 Prinzipien der KADS–Methodologie

Der KADS–Ansatz kann durch zwei Hauptprinzipien, die dem Konstruktionsprozeß für WBS zugrunde liegen, charakterisiert werden:

- das Prinzip der Zwischenmodelle (Multiple oder auch Intermediate Models) und
- das Prinzip der Modellierung auf dem Knowledge Level.

Das zweite Prinzip erlaubt, das Problemlösungswissen auf einer implementationsunabhängigen Ebene zu beschreiben. Grundlage hierfür bilden die Ebenenmodelle der Wissensrepräsentation (siehe [Newell, 1982]).
Der Knowledge Level ist dadurch charakterisiert, daß er keine Struktur besitzt. Bei der Übertragung in den Symbol Level muß diese erst hinzugefügt werden. Mit dem Knowledge Level kann das rationale Verhalten eines Systems beschrieben werden, d.h. es kann angegeben werden, warum das System (der Agent) eine bestimmte Handlung ausführt, unabhängig von der symbolischen Repräsentation (Regeln, Frames, etc.), also dem Symbol Level. Das Ziel der Einführung des Knowledge Level in KADS ist, die Organisation des Wissens im System zu verdeutlichen, d.h. eine implementationsunabhängige Beschreibung der Rollen zu liefern, die die unterschiedlichen Wissenselemente im Problemlösungsprozeß spielen.
Das Prinzip der Zwischenmodelle dient dazu, die Komplexität des Knowledge Engineering Prozesses in Form einer Divide– and Conquer–Strategie zu reduzieren und damit handhabbar zu machen. Jedes Modell beschreibt einen Aspekt des zu erstellenden Systems und kann so von den anderen Aspekten abstrahieren. Die folgenden Modelle werden im KADS–Ansatz erstellt:

- das Organisationsmodell,
- das Anwendungsmodell,
- das Aufgabenmodell,
- das Expertisenmodell,
- das Kooperationsmodell,
- das konzeptuelle Modell und
- das Designmodell.

2.1.3 Die KADS–Modelle

Der KADS–Ansatz umfaßt die Konstruktion der oben aufgelisteten Zwischenmodelle, die gemäß Abbildung 2.4 hierarchisch angeordnet werden können. Die Verbindungen deuten dabei an, daß Informationen von einem Modell in die Konstruktion des nächsten eingehen. Der folgende Abschnitt bietet eine kurze Übersicht über die einzelnen Modelle.

Das Organisationsmodell

Das Organisationsmodell liefert eine Analyse des Umfelds, in dem das WBS eingesetzt wird. Dazu gehört die Beschreibung der Funktionen, Aufgaben und Engpässe in diesem Umfeld. Die

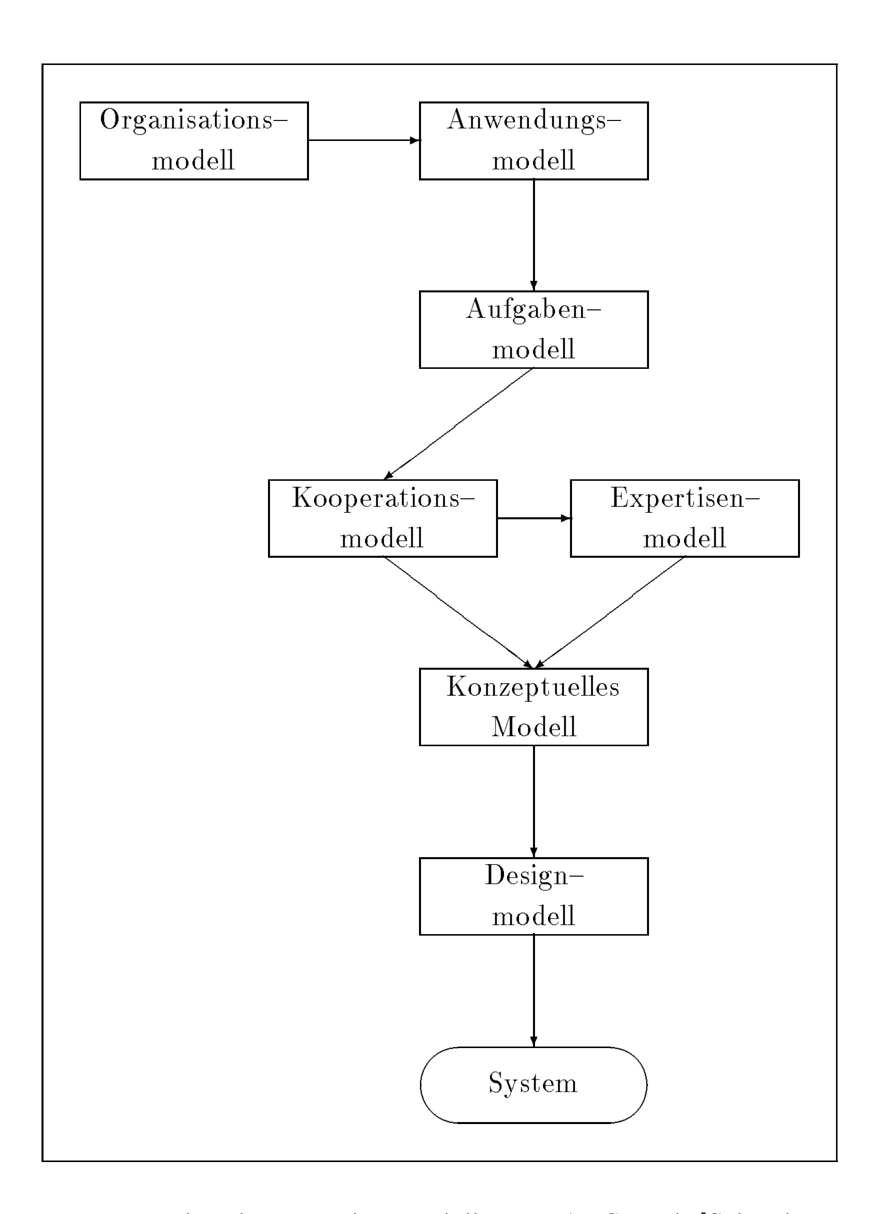


Abbildung 2.4: Hierarchie der Zwischenmodelle in KADS nach [Schreiber et al, 1993]

Konstruktion eines WBS kann zu Veränderungen in der Organisation führen. Vorhersagen, wie die Einführung des WBS die Organisation verändern wird (Feedback und Technology Assessment), gehören daher zum Organisationsmodell.

Das Anwendungsmodell

Das Anwendungsmodell umfaßt eine Problem– und Funktionsanalyse. Es definiert das Problem, das durch den Einsatz des Systems gelöst werden soll, und beschreibt die Funktionen des Systems in der Organisation.
Außerdem werden die externen Zwänge, die bei der Entwicklung des Systems berücksichtigt werden müssen, dargestellt. Unter diesen Rahmenbedingungen werden beispielsweise Anforderungen an die Geschwindigkeit und Effizienz oder bestimmte Hard– und Software verstanden.

Aufgabenmodell

Im Aufgabenmodell wird festgelegt, wie die Funktion des Systems, die im Anwendungsmodell spezifiziert wurde, durch die Ausführung einer Reihe von Tasks (Aufgaben) wahrgenommen werden kann. Es wird daher eine Relation zwischen einer Funktion und mehreren Tasks hergestellt. Dabei kann eine Funktion (ein Ziel) auf mehreren alternativen Wegen erreicht werden. Welche Alternative die günstigste ist, hängt von den Charakteristiken der Anwendung, der Verfügbarkeit von Wissen, den Erfordernissen des Benutzers und externen Funktionen ab. Das Aufgabenmodell beinhaltet

- eine Aufgabenzerlegung (Task Decomposition) und
- eine Aufgabenverteilung (Task Distribution).

Das Expertisenmodell

Der Bau des Expertisenmodells ist die zentrale Aktivität im Prozeß der Konstruktion eines WBS. Hier zeigt sich der Unterschied zwischen der Entwicklung eines WBS und der Entwicklung konventioneller Softwaresysteme.
Dieses Modell spezifiziert die Problemlösungsexpertise, die erforderlich ist, um die dem System zugeordneten Problemlösungsaufgaben durchzuführen. Das Expertisenmodell ist ein Knowledge Level Modell, bei dem das angestrebete Verhalten des Systems und die für dieses Verhalten erforderlichen Wissenstypen im Vordergrund stehen. Von Implementationsdetails, die die konkrete Realisierung festlegen, wird abstrahiert. Das Modell erfüllt somit die Rolle einer funktionalen Spezifikation des Problemlösungsteils des WBS.
Das Expertisenmodell geht nicht weiter auf die Aufgaben ein, die die Kooperation von Benutzer und System erfordern.

Kooperationsmodell

Das Kooperationsmodell enthält eine Spezifikation der Funktionalität der im Aufgabenmodell beschriebenen Unteraufgaben, die eine Kommunikation zwischen dem Benutzer und dem System, also den Agenten, erfordern. Diese Tasks werden auch Tranfer Tasks genannt, da sie Informationen zwischen Benutzer und System übertragen. Die Verteilung der Aufgaben auf die Agenten resultiert in einem Modell kooperativer Problemlösung, bei dem Benutzer und System zusammen ein Ziel erreichen.

Das konzeptuelle Modell = Expertisenmodell + Kooperationsmodell

Das Kooperations– und Expertisenmodell liefern zusammen eine implementationsunabhängige Spezifikation des Verhaltens des zu bildenden Systems. Das konzeptuelle Modell stellt eine abstrakte Beschreibung der Objekte und Operationen des Systems zur Verfügung, die in einer natürlichen Sprache ausgedrückt wird. Damit kann das konzeptuelle Modell als Grundlage für die Kommunikation zwischen Experte und Wissensingenieur dienen und verbindet Phänomene der realen Welt mit dem kognitiven Modell des Beobachters.

Das Designmodell

Das Designmodell beschreibt die Wissensrepräsentationstechniken, die verwendet werden, um die Elemente des konzeptuellen Modells zu implementieren und damit das spezifizierte Verhalten zu realisieren. Es ist also ein Modell auf dem Symbol Level.
Hier müssen auch die im Anwendungsmodell beschriebenen Anforderungen an Geschwindigkeit sowie an Hard– und Software berücksichtigt werden.
Der Vorteil der Trennung in konzeptuelles Modell und Designmodell liegt darin, daß der Wissensingenieur bei der Entwicklung des konzeptuellen Modells nicht auf den Repräsentationsformalismus achten muß. Der Rahmen für die Modellierung der Expertise ist mehr oder weniger universell. Ein weiterer Vorteil ist, daß die Spezifikation im Expertisenmodell bzw. im konzeptuellen Modell für den Experten verständlicher ist als ein Symbol Level Modell.
Dennoch sind das konzeptuelle Modell und das Designmodell nicht unabhängig voneinander. Trotz der Trennung muß die Spezifikation später auf einen Implementationslevel übertragen werden.
Die Abbildung 2.5 verdeutlicht die verschiedenen Rollen, die das konzeptuelle Modell und das Designmodell im Knowledge Engineering Prozeß spielen. Der Wissensingenieur konstruiert ein konzeptuelles Modell auf dem Knowledge Level, indem er vom Problemlösungsverhalten des Experten abstrahiert. Durch die Verwendung eines Interpretationsrahmens in Form von generischen Modellen, Aufgabenklassen und Aufgabendomains kann dieser Prozeß un-

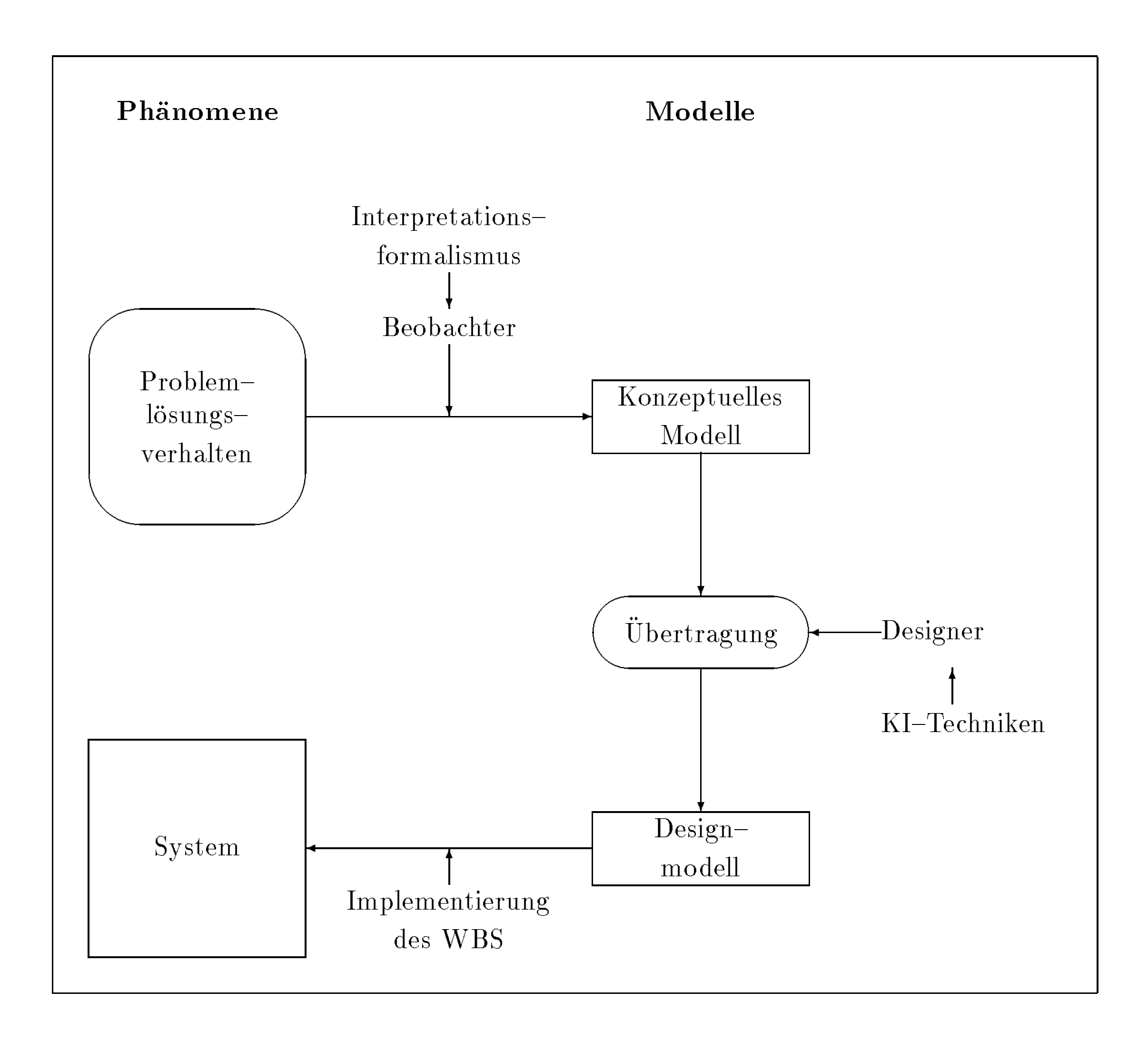


Abbildung 2.5: Rollen des Konzeptuellen Modells und des Designmodells im WBS Design nach [Schreiber et al, 1993]

terstützt werden. Ein Designer überträgt die Spezifikation des konzeptuellen Modells auf den Symbol Level, indem er durch die Auswahl geeigneter KI–Techniken und Repräsentationen das Designmodell erzeugt.

2.1.4 Modellierung der Expertise

Die Modellierung der Expertise ist der wichtigste Schritt in der Konstruktion eines WBS und hebt den Prozeß von der Entwicklung konventioneller Systeme ab. Aus diesem Grund wird die Modellierung der Expertise im folgenden näher erläutert.
Der KADS–Ansatz geht davon aus, daß verschiedene generische Wissensarten unterschieden werden können. Dabei wird berücksichtigt, welche Rolle sie im Problemlösungsprozeß über-

nehmen. Diese Wissenstypen können auf verschiedenen Ebenen dargestellt werden, die nur eine begrenzte Interaktion haben.
Eine Unterscheidung zwischen dem Sachbereichswissen (Domain Knowledge) und dem Kontrollwissen wird häufig vorgenommen. In KADS erfolgt aber eine weitergehende Einteilung nach epistemischen Gesichtspunkten.
Danach gibt es folgende Aufteilung:

1. Sachbereichswissen (Domain Knowledge): Es besteht aus statischem Wissen, das eine deklarative Theorie des Sachbereichs beschreibt.

2. Inferenzwissen (Inference Knowledge): Es handelt sich um die erste Art von Kontrollwissen, das primitive Inferenzen spezifiziert, die im Sachbereich vorkommen.

3. Aufgabenwissen (Task Knowledge): Dieses Wissen repräsentiert die elementaren Aufgaben (Problemlösungsmethoden), die aus den primitiven Inferenzen zusammengesetzt werden.

4. Strategisches Wissen (Strategic Knowledge): Die dritte Art des Kontrollwissens identifiziert die Ziele, die zur Lösung eines Problems erreicht werden müssen.

Die vier Wissensarten werden im folgenden detailliert erläutert.

Sachbereichswissen

Das Sachbereichswissen bietet eine Konzeptualisierung des Sachbereichs in Form einer deklarativen Sachbereichstheorie. Die verwendeten Elemente für diese Beschreibung basieren auf den primitiven epistemiologischen Konstrukten aus KL–ONE (siehe [Brachman et al, 1985]). Danach können folgende Elemente des Sachbereichswissens unterschieden werden:

Konzepte: Sie sind die zentralen Objekte des Sachbereichswissens.

Eigenschaft / Wert: Die Konzepte haben Eigenschaften, die durch ihren Namen und die möglichen Werte definiert werden.

Relationen zwischen Konzepten: Häufig vorkommende Relationen sind z.B. die Sub–Class oder die Part–Of Relationen.

Relationen zwischen Aussagen über Eigenschaften bzw. Eigenschaftswerte: Damit sind beispielsweise kausale oder zeitliche Zusammenhänge gemeint.

Struktur: Sie wird benutzt zur Repräsentation eines komplexen Objekts, das aus mehreren Konzepten und Relationen besteht.

Durch die Verwendung dieser Konstrukte wird ein Domain Schema für den Sachbereich entworfen. Dieses Schema beschreibt die Struktur der Aussagen in der Sachbereichstheorie. Es spezifiziert die wichtigsten Schritte in der Konzeptualisierung des Sachbereichs.
Das Sachbereichswissen sollte unabhängig von der Aufgabe sein, so daß es von unterschiedlichen Problemlösungshandlungen benutzt werden kann.

Inferenzwissen

Das Inferenzwissen spezifiziert primitive Inferenzen, die durch den Namen, eine Input/Output–Spezifikation und das verwendete Sachbereichswissen definiert werden. Das Verhalten der Inferenz läßt sich durch Newells Rationalitätsprinzip (siehe [Newell, 1982]) bestimmen:

> Wenn ein Agent weiß, daß eine Aktion, die ausgeführt werden kann, ein gewünschtes Ziel erreicht, dann wählt der Agent diese Aktion aus.

Im Expertisenmodell von KADS werden folgende Terme verwendet, um primitive Inferenzen zu beschreiben:

Wissensquellen (Knowledge Sources): Sie führen den eigentlichen Inferenzschritt aus. Aus den Input–Daten kann unter Verwendung des Sachbereichswissens neue Information (Wissen) hergeleitet werden.

Metaklasse (Meta–Class): Die Datenelemente, auf die die Knowledge Sources zurückgreifen und die sie produzieren (d.h. Input und Output), werden als Metaklassen bezeichnet.

Metaklassen fungieren als Platzhalter für Objekte des Sachbereichs. Sie beschreiben die Rolle dieser Objekte im Problemlösungsprozeß.

Die Zuordnung von Sachbereichsobjekten zu Metaklassen ist nicht eindeutig, d.h. Objekte können mehr als einer Metaklasse angehören. Jede Metaklasse kann sowohl eine Input–Metaklasse für eine Wissensquelle als auch eine Output–Metaklasse darstellen.

Sachbereichssicht (Domain View): Der Domain View spezifiziert die Referenz der Metaklassen und der Wissensquellen zum Sachbereichswissen, d.h. es wird beschrieben, welche Teile des Domain Knowledge an welcher Stelle des Inferenzprozesses verwendet werden.

Durch die Trennung von Sachbereichs– und Inferenzwissen kann dasselbe Sachbereichswissen für viele verschiedene Typen von Inferenzen verwendet werden.
Die Menge aller primitiven Inferenzen kann grafisch zu einer Inferenzstruktur zusammengefaßt werden, die die Problemlösungskompetenz des WBS wiedergibt. Die Reihenfolge, in der die

Inferenzen ausgeführt werden, wird hier noch nicht spezifiziert. Dies erfolgt erst durch das Aufgabenwissen.

Aufgabenwissen

Durch das Aufgabenwissen wird bestimmt, wie primitive Inferenzen kombiniert werden können, um ein vorgegebenes Ziel zu erreichen. Folgende Konstrukte werden zur Beschreibung des Aufgabenwissens verwendet:

Aufgabe (Task): Der primäre Wissenstyp dieser Kategorie kann als zusammengesetzte Problemlösungsmethode bezeichnet werden. Die Aufgabe wird in eine Menge von Unteraufgaben zerlegt.

Kontrollterme (Control Terms): Kontrollterme sind Bezeichnungen für eine Menge von Metaklassen–Elementen.

Aufgabenstruktur (Task Structure): Die Aufgabenstruktur legt den Ablauf des Problemlösungsprozesses fest und greift dabei auf die Unteraufgaben zurück. Es gibt drei Arten von Unteraufgaben:

1. Primitive Problemlösungsaufgaben, bestehend aus Inferenzen, die im Inferenzwissen spezifiziert wurden.
2. Zusammengesetzte Problemlösungsaufgaben, die aus Aufgaben bestehen, die im Aufgabenwissen beschrieben wurden.
3. Transferaufgaben, die eine Interaktion mit dem Benutzer des Systems erfordern. Diese werden hier nur als Black Boxes spezifiziert.

Eine wichtige Eigenschaft des Aufgabenwissens ist, daß hier keine Bezüge zum Sachbereichswissen, sondern nur zum Inferenzwissen vorkommen.
Die Zerlegung der Aufgaben in Unteraufgaben kann in Form eines Aufgabenbaums dargestellt werden.

Strategisches Wissen

Dieses Wissen bestimmt, welche Ziele relevant sind, um ein Problem zu lösen. Wie die Ziele erreicht werden, ist durch das Aufgabenwissen festgelegt.
Treten im Problemlösungsprozeß Widersprüche, Konflikte oder unvorhergesehene Ereignisse ein, so muß das strategische Wissen Vorschläge zur weiteren Vorgehensweise, ggf. durch neue Annahmen, unterbreiten.

2.1.5 Modellierung der Kooperation

Neben der Modellierung der Expertise ist auch die Modellierung der Benutzer–System–Kooperation von zentraler Bedeutung. Die Erkenntnisse, die durch diese Modellierung gewonnen werden, fließen mit in das Expertisenmodell ein und münden direkt in der Erstellung des Kooperations– und des Aufgabenmodells.
Die Kooperation basiert dabei auf drei wesentlichen Elementen:

Verteilung der Aufgaben: Die Verteilung der Aufgaben besteht in einer Aufteilung der Aufgaben in Unteraufgaben, die dann an die verschiedenen Agenten verteilt werden können. Auf diese Weise muß die Gesamtaufgabe nicht durch einen Agenten alleine ausgeführt werden. Es kann vorkommen, daß ihm das Ziel dieser Gesamtaufgabe unbekannt ist. Er wird nur mit der Durchführung einer Unteraufgabe betraut.

Abhängigkeiten: Durch die Aufgabenverteilung entstehen Abhängigkeiten zwischen Unteraufgaben in der Form, daß der Output einer Aufgabe als Input für eine andere benötigt wird. Die Input– bzw. Output–Objekte, die sogenannten Ingredients, werden zwischen den Aufgaben transferiert. Hieraus ergibt sich die Notwendigkeit zur Kommunikation zwischen den Agenten. Die Art und Weise, auf die diese Kommunikation durchgeführt wird, gehört nicht mehr zur Spezifikation der Kooperation.

Kontrolle: Agenten, an die Aufgaben im Rahmen der Task Distribution verteilt worden sind, müssen wissen, welche Unteraufgaben sie wann durchzuführen haben. Eine Überwachung der Mensch–Maschine–Kooperation durch das System ist aber nur schwer zu realisieren.

Eine feste Aufgabenverteilung, die Spezifikation der Abhängigkeiten und die Bestimmung der Kontrolle bilden das Kooperationsmodell.
Die Rolle der Kooperationsanalyse ist in Abbildung 2.6 dargestellt. Der Ausgangspunkt für die Konstruktion eines WBS ist oft ein Konzept für eine „intelligente Automatisierung“ bestimmter Funktionen. In der vorhandenen Aufteilung der Aufgaben sind diese Funktionen häufig verbunden mit den Engpässen, den Bottlenecks der Organisation.
Im folgenden werden das Aufgaben– und das Kooperationsmodell vorgestellt.

Aufgabenmodell: Aufteilung und Verteilung der Aufgaben

Im Aufgabenmodell werden die Aufgaben in kleinere Unteraufgaben zerlegt und dann an die Agenten verteilt.
Zur Durchführung der Task Decomposition werden von den KADS–Autoren folgende Heuristiken vorgeschlagen:

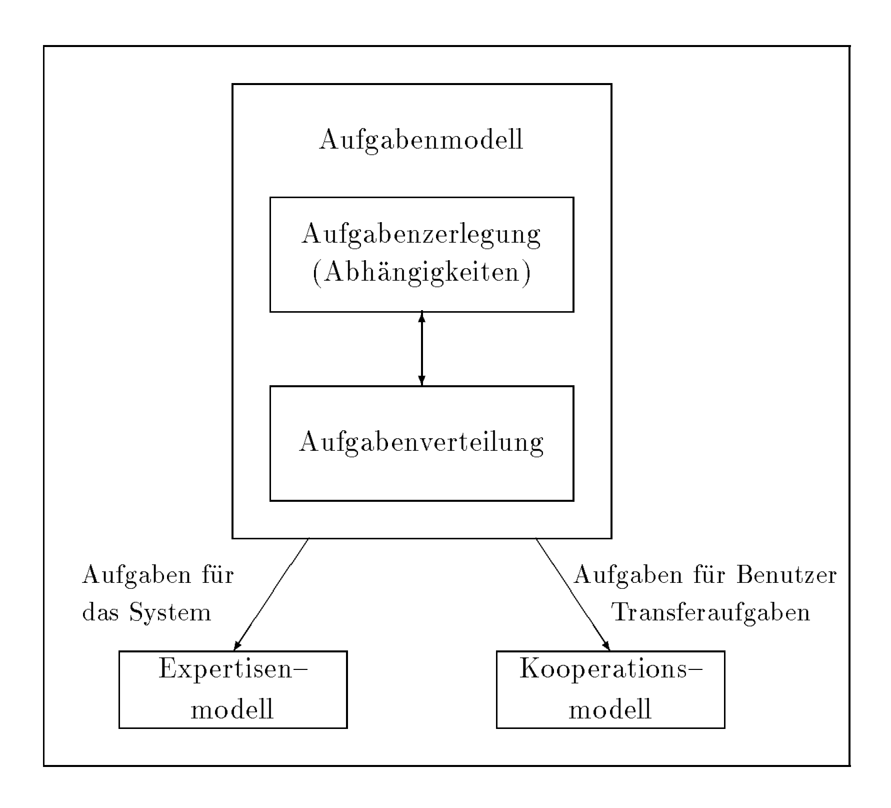


Abbildung 2.6: Rollen der Kooperationsanalyse nach [Schreiber et al, 1993]

Objektzerlegung (Object Decomposition): Kann der Output eines Tasks in mehrere Teile zerlegt werden, so können diese Teile von unterschiedlichen Unteraufgaben produziert werden. Diese Aufteilung wird auch als Zielzerlegung bezeichnet.

Eine effiziente Aufgabenzerlegung zeigt sich in der Anzahl der Beziehungen zwischen den Unteraufgaben. Je weniger Verbindungen bestehen, desto weniger Kooperation und damit Kommunikationsaufwand ist erforderlich.

Objektverfeinerung (Object Refinement): Sind in dem Output einer Aufgabe unterschiedliche Abstraktionsstufen festzustellen, so können mögliche Unteraufgaben aus einer Reihe von Verfeinerungsschritten bestehen[3].

Funktionale Sequenzierung (Funktional Sequencing): Die funktionale Sequenzierung zerlegt die Aufgaben nach funktionalen Kriterien in eine Reihe von Operationen und Transformationen. Diese operieren dabei immer auf demselben Objekt, d.h. die Bearbeitung eines Objekts wird in mehrere Schritte unterteilt.

Wissenstypisierung (Knowledge Typing): Können verschiedene Wissenstypen unterschieden werden, die zur Ausführung einer Aufgabe benötigt werden, so liegt hier eine Modularisierung der Aufgabe bzgl. dieser Typen auf der Hand.

Die Spezifikation der Aufgabenzerlegung kann in einem UND/ODER–Grafen dargestellt werden, wobei die ODER–Zweige verschiedene Methoden der Zerlegung darstellen. Werden Abhängigkeiten zwischen den Unteraufgaben berücksichtigt, so können hierarchische Datenflußdiagramme verwendet werden. Die Pfeile, die den Fluß der Daten andeuten, werden dabei mit dem Objekt (Ingredient) beschriftet. Gibt es eine Beziehung zwischen zwei Tasks, so stehen alle übergeordneten Tasks ebenfalls in Beziehung.
Im Anschluß an die Aufgabenzerlegung erfolgt die Verteilung der Unteraufgaben an die Agenten. Es gibt eine Reihe von Kriterien für die Verteilung der Aufgaben. Das Ziel der Aufgabenverteilung ist die optimale Nutzung der Kompetenz, d.h. des Wissens und der Fähigkeiten der Agenten. Daher ist diese das Hauptkriterium für die Verteilung. Um die Kompetenz beurteilen zu können, müssen Annahmen über die Fähigkeiten der zukünftigen Benutzer gemacht werden. Dabei sind die Benutzer in Gruppen im Sinne von Typen mit homogenen Fähigkeiten einzuteilen, für die unterschiedliche Aufgabenverteilungen realisiert werden müssen. Die Verteilung der Aufgaben zwischen Mensch und System ist in der Regel einfacher. Um die speziellen Fähigkeiten von Mensch und System zu nutzen, werden die Aufgaben, die den „gesunden Menschenverstand“ benötigen, dem Benutzer zugeordnet, während Aufgaben, die ein

[3] Hier zeigt sich auch die top–down–Vorgehensweise in der Aufgabenzerlegung.

hohes Maß an Informationsmanagement, Wiederholung oder bekannten Routinen beinhalten, dem System zugeordnet werden.
Wird eine Unteraufgabe sowohl dem Benutzer als auch dem System zugeordnet, so kommen dafür mehrere Gründe in Betracht:

- die Unterteilung ist nicht fein genug (Insufficient Refinement),
- es gibt alternative Verteilungen der Aufgaben,
- es findet eine parallele Ausführung von Aufgaben statt, um die Ergebnisse vergleichen zu können, oder
- ein Agent gibt Instruktionen an einen ausführenden Agenten.

Ist die Verteilung der Aufgaben abgeschlossen, so muß für alle Aufgaben, die dem System zugeordnet worden sind, entschieden werden, ob sie mit konventionellen Ansätzen bearbeitet werden können oder ob sie „wissensintensiv“ sind. Im letzteren Fall sollten Knowledge Enigineering–Techniken zum Einsatz kommen. Dann können diese Aufgaben mithilfe der in 2.1.4 beschriebenen Modellierung der Expertise analysiert werden.

Kooperationsmodell

Im Kooperationsmodell wird die Kooperation und Kommunikation mit dem Benutzer spezifiziert. Die Problemlösung, die im Expertisenmodell festgelegt ist, und die Kooperation/Kommunikation werden durch einen strategischen Meta–Level kontrolliert, der ebenfalls im Kooperationsmodell festgelegt wird. Das wichtigste Element dieser Kontrolle ist die „Initiative“. Sie gibt an, welcher Agent die Kommunikation auslösen kann.
Abhängigkeiten zwischen Unteraufgaben im Task Modell weisen darauf hin, daß hier Kooperation bzw. Kommunikation erforderlich ist und damit Objekte transferiert werden müssen. Diese Objekte, die Ingredients, werden von ihren Besitzern produziert. Das Besitzen eines Ingredients soll hier andeuten, daß der Agent, der ein Ingredient produziert hat, die Kontrolle hierüber ausübt. Ingredients können in drei Typen unterteilt werden:

- Informationen: Daten, Aussagen über Probleme, Ansichten, Auswertungen, Geschichten, Zustände etc.,
- Wissen: generisches Wissen, das zur Erklärung benutzt werden kann, und
- Fähigkeiten: Instruktionen, die angeben, wie eine Aufgabe auszuführen ist.

Die Kombination von Ingredients, der Initiative und der Kontrolle über Ingredients führt zu vier unterschiedlichen Typen von Transferaufgaben, die in Abbildung 2.7 dargestellt sind.

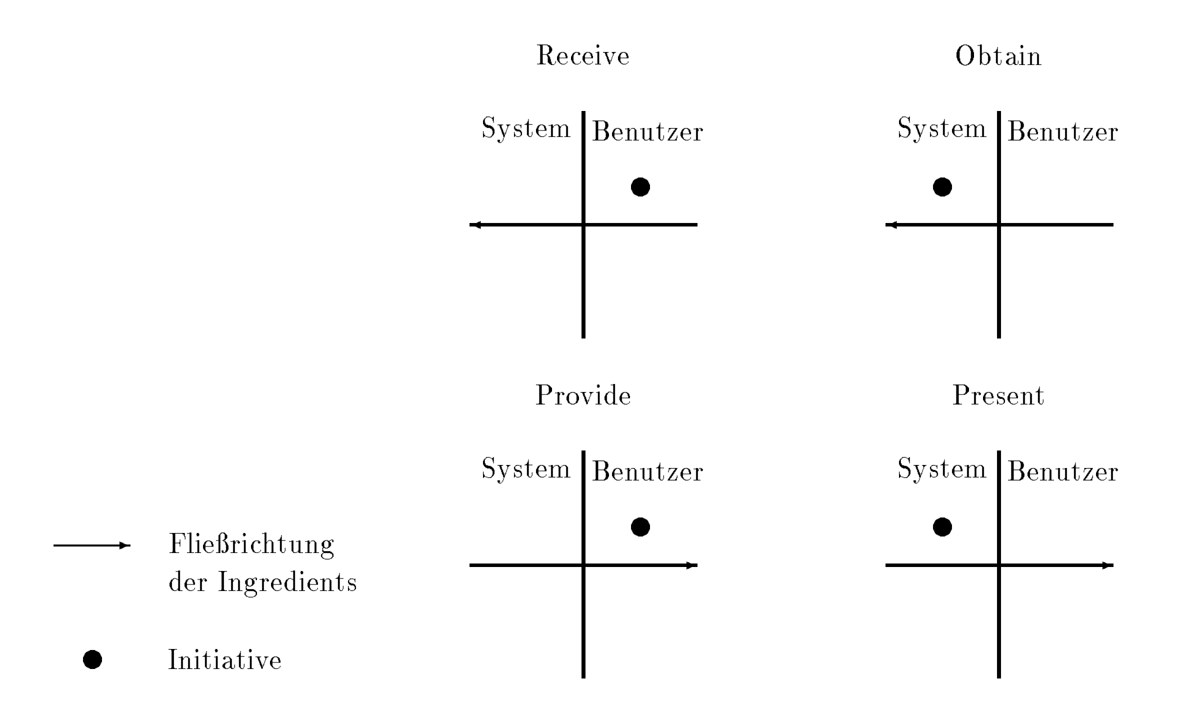


Abbildung 2.7: Vier Typen von Transferaufgaben in KADS [Schreiber et al, 1993]

Zusätzlich gibt es den Transfer Task „Negotiate", der Informationen über die Kooperation oder die Problemlösung selbst transferiert.
Um das Kooperationsmodell zu testen, kann ein Prototyp des Systems erstellt werden oder das Verhalten durch ein sogenanntes Wizard–of–Oz Experiment überprüft werden, siehe [Schreiber et al, 1993].

2.1.6 Diskussion

In diesem Abschnitt werden einige Vor– und Nachteile des KADS–Ansatzes aufgezeigt, die sich während der Arbeit mit dieser Methodik ergeben haben.
Als positiv ist zu bewerten, daß die Wissensrepräsentation im Rahmen von KADS nicht mehr als Transfer eines fertigen Modells vom Experten zum WBS betrachtet wird. Es wird ein vollständiges, funktionales Knowledge Level Modell für das Problemlösungsverhalten erstellt. Dabei kann der gesamte Entwurfsprozeß mithilfe der KADS–Methodik durchgeführt werden.
Problematisch ist dagegen die Phaseneinteilung, siehe Abbildung 2.8. Es wird das Ziel verfolgt, die Revisionen im Konstruktionsprozeß zu minimieren, indem eine Linearität gemäß einer Wasserfallstrategie angestrebt wird. Diese Phasentrennung erwies sich in der Konstruktion des im Rahmen dieser Arbeit erstellten WBS als eher ungünstig.
Mit dem KADS–Ansatz können notwendige Revisionen nicht systemgestützt durchgeführt

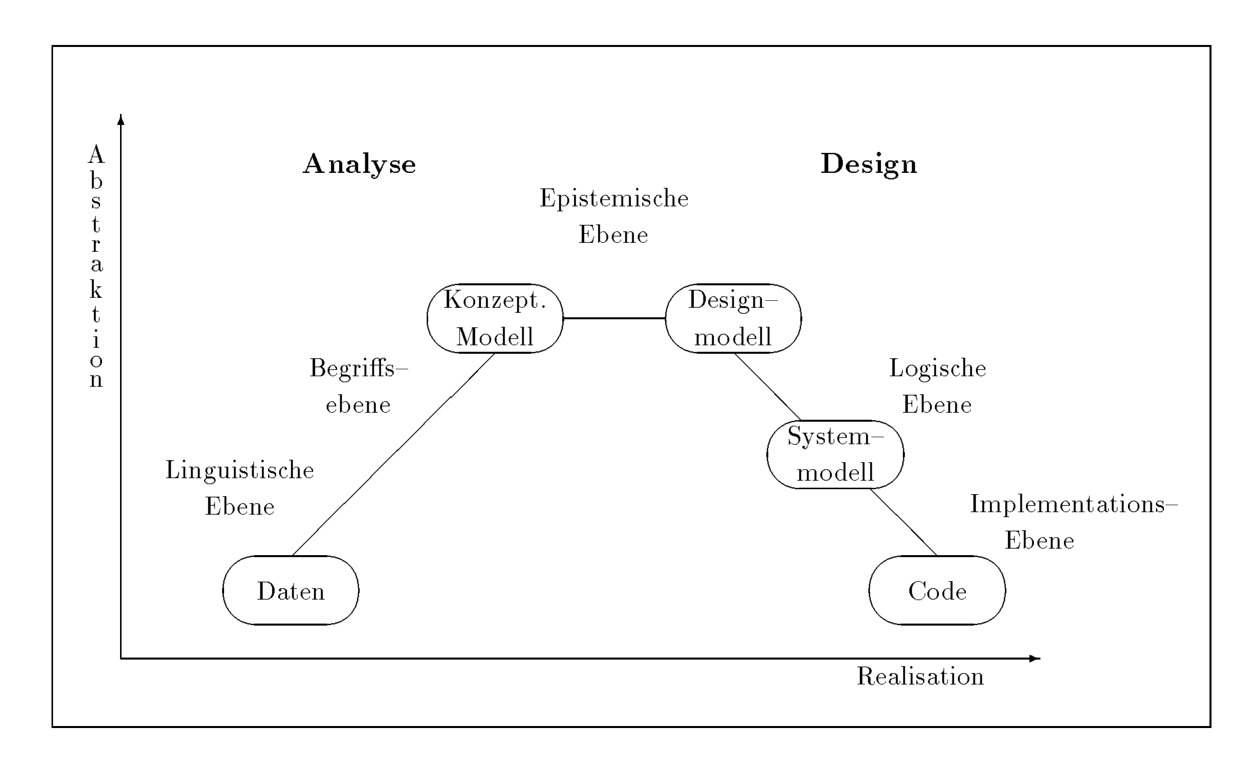


Abbildung 2.8: Phasen und Ebenen in KADS [Schreiber et al, 1993]

werden. Der Grund hierfür liegt darin, daß das konzeptuelle Modell nicht operational ist, d.h. das Verhalten kann nicht „vorgezeigt" werden. Dieser Nachteil des KADS–Ansatzes hat sich hier als besonders gravierend herausgestellt. Ein funktionaler Prototyp des WBS, der mithilfe der Systeme MOBAL und Delphi erstellt werden konnte, hat sich als bessere Basis für die Kommunikation mit den Experten erwiesen, als das konzeptuelle KADS–Modell.

Desweitern ist aufgefallen, daß der Schritt vom konzeptuellen Modell zum Designmodell sehr schwer fällt. Es kann schlecht festgestellt werden, ob das Knowledge Level Modell tatsächlich durch das operationale Modell realisiert wird. Dadurch, daß das konzeptuelle Modell näher an die Begriffswelt des Experten gerückt ist, wird der Abstand zum Designmodell größer. Die Sprachkonstrukte der Knowledge Level Modelle müssen weitesgehend intuitiv interpretiert werden, da eine formale Grundlage fehlt. Die Vorschläge, die für das Structure Preserving Design (siehe [Schreiber et al, 1993], Seite 124 – 130) gemacht werden, konnten in dieser Arbeit nur teilweise umgesetzt werden.

2.2 MOBAL

MOBAL (siehe [Morik et al, 1993]) ist ein System zur Wissensmodellierung, das auf den Prinzipien der oben erwähnten Sloppy Modeling Methodik basiert. Kennzeichnend ist die Verbindung von systemunterstützter Modellierung und maschinellem Lernen. Der Aufbau des Modells kann in einer Bottom–Up–Vorgehensweise erfolgen: zunächst gibt der Knowledge Engineer konkrete Beobachtungen in Form von Beispielen in das System ein. Durch eine integrierte Lernkomponente können aus diesen Beispielen generellere Definitionen und Zusammenhänge in Form von Regeln gelernt werden. Ein ungenaues, unvollständiges Modell kann so inkrementell verbessert werden.

Die Kooperation zwischen Wissensingenieur und System folgt dabei dem Balanced Cooperative Modeling Prinzip, d.h. die Aufgabenverteilung ist sehr flexibel. Alle Aufgaben, die vom Wissensingenieur durchgeführt werden, können auch vom System übernommen werden und umgekehrt.

Eine wichtige Eigenschaft des Systems ist die Revidierbarkeit des erstellten Modells. Im Gegensatz zum KADS–Ansatz handelt es sich hier um ein operationales Modell, so daß der Revisionsprozeß durch das System unterstützt werden kann. Das Verhalten des Modells kann zu jedem Zeitpunkt überprüft werden. Die Konsequenzen von Veränderungen am Modell, beispielsweise Inkonsistenzen, werden vom System entdeckt, aufgezeigt und können mit Unterstützung des Systems rückgängig gemacht bzw. behoben werden.

MOBAL integriert eine Reihe verschiedener Tools, die den Benutzer bei der Modellierung unterstützen. Entscheidend ist hier, daß alle Tools die gleiche Repräsentation verwenden.

Im folgenden Abschnitt wird diese Wissensrepräsentation vorgestellt. Anschließend wird das Tool von MOBAL beschrieben, das im Rahmen dieser Arbeit vorwiegend verwendet wurde. Dabei handelt es sich um das in das System integrierte Lernverfahren RDT (Rule Discovery Tool), das in einer vorhandenen Wissensbasis Regeln entdecken kann. Mithilfe eines weiteren Tools, nämlich des Rule Restructuring Tools (RRT), das im darauffolgenden Abschnitt vorgestellt wird, kann eine gelernte oder eingegebene Regelmenge analysiert und gegebenfalls neu strukturiert werden.

2.2.1 MOBALs Wissensrepräsentation

Die Wissensrepräsentation von MOBAL baut auf einer sortierten Hornklausellogik erster Stufe auf, die allerdings negierte Literale sowohl in den Prämissen als auch der Konklusion erlaubt. Zur Beschränkung der Komplexität wird die Verwendung von Funktionssymbolen ausgeschlossen.

Das Sachbereichswissen wird durch Fakten, Regeln, Regelmodelle (Metaprädikate) und wei-

`eq(_,_)`	equal/unifiable (equal)
`ne(_,_)`	not equal
`lt(_,_)`	lower than
`gt(_,_)`	greater than
`ge(_,_)`	greater equal
`le(_,_)`	lower equal
`add(_,_,_)`	addition
`sub(_,_,_)`	subtraction
`prod(_,_,_)`	mulitplication
`div(_,_,_)`	division

Tabelle 2.1: Built–In Prädikate

teres Metawissen sowie durch Sorten und eine Prädikattopologie dargestellt. Diese Wissenselemente werden nun genauer erläutert.

Fakten

Fakten bilden die Grundlage einer Wissensbasis. Durch sie können Relationen, Eigenschaften und Konzeptzugehörigkeiten (Begriffe) ausgedrückt werden. Fakten können sowohl vom Benutzer eingegeben, als auch durch die Inferenzmaschine (siehe [Morik et al, 1993]) abgeleitet werden.
Ein Faktum besteht aus einem n–stelligen Prädikat, das auf konstante Terme angewendet wird:

$$pred(Term_1, \ldots, Term_n)$$

Es werden vier Wahrheitwerte verwendet: `True, False, Unknown` und `Contradictory`. Fakten können vom Benutzer negiert oder nicht negiert eingegeben werden. Die Fakten, die dem System (noch) nicht zur Verfügung stehen, werden als `Unknown` interpretiert, d.h. es wird keine `Closed World Assumption` getroffen, da diese einer inkrementellen Erstellung des Modells im Wege steht. Treten in der Wissensbasis Fakten in negierter und nicht negierter Form gleichzeitig auf, so haben sie den Status `Contradictory`.
Die erforderlichen Prädikate können durch den Benutzer deklariert werden (siehe Abschnitt Sorten). Zusätzlich werden in MOBAL auch sogenannte „Built–In"–Prädikate und „Autoepistemische" Operatoren zur Verfügung gestellt, siehe Tabellen 2.1 und 2.2.

```
max_of(!<variable>,!<proposition>,?<maximum>)
min_of(!<variable>,!<proposition>,?<minimum>)
sum_of(!<variable>,!<proposition>,?<sum>)
count(!<proposition>,?<number>)
unknown(!<proposition>)
```

Tabelle 2.2: Autoepistemische Operatoren

Regeln

Regeln beschreiben Begriffs– und Eigenschaftsrelationen sowie notwendige und hinreichende Bedingungen für Konzepte. Mithilfe der Inferenzmaschine werden Fakten aus den Regeln inferiert, d.h. die Regeln leiten aus dem Sachbereichswissen neues Wissen ab.
Regeln können durch den Benutzer eingegeben, aber auch mittels RDT oder anderen Tools des Systems gelernt werden.
Regeln (Klauseln) bestehen aus genau einer Konklusion (Kopf der Klausel) und m Prämissen (Rumpf), $m \epsilon I\!N$:

$$pred_1(Term_1^{(1)}, \ldots, Term_{n_1}^{(1)}) \ \& \ldots \& \ pred_m(Term_{n_1}^{(m)}, \ldots, Term_{n_m}^{(m)})$$

$$\rightarrow pred_{concl}(Term_1^{(concl)}, \ldots, Term_{n_{concl}}^{(concl)}).$$

Sowohl die Konklusion als auch die Prämissen sind Literale: sie wenden ähnlich wie die Fakten jeweils ein Prädikat auf eine bestimmte Anzahl von Termen an. Die Terme müssen hier allerdings nicht konstant sein, sondern dürfen auch Variablen enthalten. Variablen, die in der Konklusion der Regel vorkommen, müssen auch in den Prämissen zu finden sein, d.h. die Konklusion darf keine neuen Variablen einführen. Die Prämissen sind konjunktiv verknüpft. Jeder Regel wird ein sogenanntes Support Set zugeordnet, das festlegt, welche Terme bei Inferenzen für die Variablen eingesetzt werden dürfen. Es kann damit eine Menge von Ausnahmen für eine Regel definiert werden.

Regelmodelle

Die beiden bisher beschriebenen Wissenselemente gehören zur Prädikatenlogik erster Stufe. Regelmodelle sind einer Metaebene zuzuordnen, die über diese Stufe hinausgeht. Sie werden auch als Metaprädikate bezeichnet. Es existieren weitere Wissenselemente auf dieser Metaebene: die Metafakten und die Metaregeln. Hierauf wird in diesem Rahmen nicht weiter eingegangen (siehe dazu [Morik et al, 1993]).

Ein Regelmodell ist eine Regel, die Prädikatvariablen anstelle von Prädikaten enthält. Regelmodelle beschreiben den Hypothesenraum für das Lernen, indem sie die Struktur der zu lernenden Regeln vorgeben. Für die Generierung von Hypothesen werden die Prädikatvariablen durch Prädikate instanziiert. Dabei kann eine n–stellige Prädikatvariable durch Prädikate der gleichen Stelligkeit ersetzt werden. Ein vollständig instanziiertes Regelmodell ist eine Regel.
Ein Metaprädikat besteht aus dem eigentlichen Regelmodell und einem Header, der den Namen des Metaprädikats angibt und die verwendeten Prädikatvariablen sowie die konstanten Terme aufführt. Dazu betrachte man das folgende Beispiel:

> Man hat das Metaprädikat
>
> MP1(P1,P2,Q): P1(X,Y) & P2(X) $\rightarrow$ Q(Y).
>
> Es müssen nun drei Prädikate gefunden werden, zwei einstellige und ein zweistelliges, mit denen die Prädikatvariablen P1, P2 und Q instanziiert werden können. Die Termvariablen X und Y legen weiter fest, daß das zweite Argument von P1 in der Konklusion und in der zweiten Prämisse vorkommen muß (siehe auch Abschnitt über Sorten).

Sorten

Durch (Argument–) Sorten wird die Objektmenge des Sachbereichs in Klassen eingeteilt. In der Deklaration eines Prädikats können Sorten festgelegt werden:

pred(<Sorte_1>,...,<Sorte_n>).

Damit wird angegeben, welcher Sorte die an der jeweiligen Argumentstelle des Prädikats auftretenden Terme angehören. Sorten können auch vom System aus einer gegebenen Faktenmenge erzeugt werden.

Topologie

Während die Sorten die Objekte des Sachbereichs einteilen, strukturiert die Topologie die Prädikate. Der Benutzer kann eine Topologie erstellen, indem er die Prädikate nach semantischen Merkmalen in Gruppen (Topologieknoten) einteilt.
Aus dieser Einteilung ergibt sich, ebenso wie aus der Sorteneinteilung, eine Beschränkung des durch die Regelmodelle aufgespannten Hypothesenraums von RDT: es dürfen nur topologie– und sortenverträgliche Prädikate in die Prädikatvariablen eingesetzt werden.

2.2.2 Das Rule Discovery Tool (RDT)

Der Lernalgorithmus RDT (siehe auch [Kietz et al, 1991]) leitet aus einer Menge von Fakten Regeln ab, die diese beschreiben bzw. Zusammenhänge aufzeigen. Die gelernten Regeln können ggf. weitere Fakten, die noch nicht eingegeben wurden, ableiten und so noch nicht beobachtete Situationen beschreiben.

Als Eingabe erhält RDT eine Menge von Fakten, Regelmodellen und Parametern (siehe unten).

Der durch die Regelmodelle aufgespannte Hypothesenraum wird von RDT durchsucht. Er besteht aus der Menge aller möglichen Instanziierungen von Regelmodellen durch Prädikatsymbole, bei denen alle Prädikatvariablen durch Prädikate der richtigen Stelligkeit ersetzt worden sind.

Aus Effizienzgründen wird eine Ordnung auf den Regelmodellen festgelegt. Das Ordnungskriterium ist eine Generalisierungsbeziehung, die auf der θ–Subsumption für Klauseln (siehe [Plotkin, 1970] und [Plotkin, 1971]) aufbaut. Diese Beziehung ist wie folgt definiert:

> Ein Regelmodell R ist genereller als ein Regelmodell R' (R $\geq_{RS}$ R'), falls es eine Substitution σ angewendet auf Termvariablen und eine Substitution Σ angewendet auf Prädikatvariablen gibt, die keine unterschiedlichen Prädikatvariablen verallgemeinert, so daß gilt: $R\sigma\Sigma \subseteq$ R'.

Durch diese Ordnung kann die Suche im Hypothesenraum beschränkt werden. RDT ist ein Top–Down Lernverfahren, d.h. die Suche nach einer zu lernenden Regel wird mit dem generellsten Regelmodell begonnen. Durch die Instanziierung der Prädikatvariablen werden Hypothesen gebildet, die dann anhand der Fakten auf ihre Richtigkeit getestet werden. Es gibt drei mögliche Testergebnisse für eine Hypothese:

1. Die Hypothese ist zu generell, d.h. sie deckt zu viele falsche oder nicht vorhandene Fakten (Beispiele) ab.

2. Die Hypothese wird akzeptiert, oder

3. sie wird verworfen, d.h. sie ist bereits zu speziell, da sie zu wenige Beispiele abdeckt.

Durch die Generalisierungsbeziehung der Regelmodelle kann die Suche im Hypothesenraum eingeschränkt werden, denn sowohl Spezialisierungen von akzeptierten als auch von nicht akzeptierten Hypothesen müssen nicht weiter betrachtet werden. Spezialisierungen von akzeptierten Hypothesen sind redundant: sie enthalten überflüssige Informationen. Spezialisierungen von verworfenen Hypothesen können nicht mehr erfüllt werden, da durch weitere

Spezialisierung die Anzahl der positiven Beispiele für die Hypothese immer kleiner wird. Nur im ersten Fall wird die Suche im Hypothesenraum weiter fortgesetzt.
Weitere Einschränkungen des Hypothesenraumes werden durch die Prädikattopologie und die Sorten vorgenommen. Darauf wird an dieser Stelle nicht weiter eingegangen (siehe dazu [Morik et al, 1993]).
Für die Erzeugung der Hypothesen werden sukzessive die Prädikatvariablen eines Regelmodells instanziiert. Auch für teilweise instanziierte Hypothesen (partielle Hypothesen) wird überprüft, ob die Hypothese durch weiteres Instanziieren, was gerade einer Spezialisierung entspricht, noch akzeptiert werden kann. Die Reihenfolge der Instanziierung der Prädikatvariablen ist von besonderer Bedeutung. Zur Festlegung der Reihenfolge wird nun der Begriff der „Relation Chain“ eingeführt. Sinnvolle Regeln zeichnen sich u.a. dadurch aus, daß alle Literale im Rumpf einer Klausel mit dem Kopf dieser Klausel in Verbindung stehen. Die Rumpfliterale hängen ebenfalls durch gemeinsame Variablen voneinander ab. Durch diese Variablenstruktur läßt sich eine Ordnung der Literale festlegen. Dazu betrachte man das folgende Beispiel:

> Man hat das Regelmodell P1(Y,X) & P2(Y) $\rightarrow$ Q(X). Die Variable X wird durch die Instanziierung von Q als erstes gebunden. Es ist nun nicht sinnvoll, zunächst nach einer Instanziierung für P2 zu suchen und dann erst die Prädikatvariable P1 zu ersetzen. Da in P1 die Kopfvariable X vorkommt, können hier nicht mehr alle zweistelligen Prädikate eingesetzt werden. Kann keine passende Instanziierung für P1 gefunden werden, kann die Suche nach einer Hypothese mit diesem Regelmodell sofort beendet werden, ohne nach einer Instanziierung für P2 zu suchen.

Aus dieser Beobachtung kann folgende Definition abgeleitet werden:

1. Eine Variable X, die in der Konklusion eines Regelmodells vorkommt, ist über die leere Relation Chain mit der Konklusion verbunden, d.h. rc(X)=$\emptyset$.

2. Eine Variable X_i, $1 \leq i \leq n$, die in der Prämisse P$(X_1, X_2, \ldots, X_n)$ auftritt, ist über die Verbindungskette rc(X_i) = P $\circ$ rc(X_j) mit der Konklusion verbunden, falls die Variable X_j, 1$\leq j \leq n$ und i $\neq$ j, über die Relation Chain rc(X_j) mit der Konklusion verbunden ist.

Der Abstand einer Variablen zum Kopf der Klausel kann berechnet werden als:

$$\delta(\text{X}) = \min(\ \{\text{Länge(rc(X))} \mid \text{rc(X) verbindet X mit der Konklusion}\ \}).$$

Eine Prämisse P ist dann zuerst zu instanziieren, wenn das Minimum aller kürzesten Relation Chains von Variablen, die in P auftreten, kleiner ist, als das Minimum aller kürzesten Relation Chains von Variablen, die in den anderen Prämissen vorkommen.

Für jedes Tool des Systems MOBAL gibt es eine Reihe von Parametern, mit denen der Benutzer die Arbeitsweise des Tools beeinflussen kann. Der wohl wichtigste Parameter für das Rule Discovery Tool ist das Akzeptanzkriterium. Sowohl für die partiellen Hypothesen als auch für die Hypothesen, die vollständig instanziiert sind, wird ein Test durchgeführt, der entscheidet, ob die Hypothese weiter instanziiert werden muß (bzw. akzeptiert werden kann, falls es sich nicht um eine partielle Hypothese handelt) oder ob sie verworfen wird. Die Entscheidung, wann eine Hypothese verworfen bzw. akzeptiert werden soll, wird vom Benutzer durch das Akzeptanzkriterium bestimmt. Dazu werden folgende Werte berechnet:

pos ist die Anzahl der Instanzen, für die die Konklusion der Hypothese erfüllt ist, wenn diese Instanzen in die Prämissen eingesetzt werden. Damit handelt es sich also um die Anzahl der positiven Beispiele für die Hypothese.

neg ist die Anzahl der Instanzen, für die die Konklusion der Hypothese negiert vorliegt, wenn die Prämissen mit ihnen instanziiert sind. neg bezeichnet damit die Anzahl der negativen Beispiele der Hypothese.

pred ist die Anzahl der Instanzen, für die die Konklusion nicht in der Wissensbasis zu finden ist, wenn die Prämissen mit ihnen instanziiert sind. Hier werden die Instanzen gezählt, die aus der Hypothese neu abgeleitet werden können.

total ist die Menge aller Instanzen für die Hypothese, also total = pos + neg + pred.

unc ist die Anzahl aller Instanzen für die Konklusion, die von der Hypothese nicht abgedeckt werden.

concl ist die Anzahl aller Instanzen für die Konklusion der Hypothese.

Das Akzeptanzkriterium kann mithilfe von logischen, arithmetischen und Vergleichsoperatoren aus den obigen Ausdrücken zusammengesetzt werden, z.B.:

pos > 4 & neg < 1 & unc < (0.9 * total).

Durch das Spezialisieren einer Hypothese wird die Anzahl der positiven, negativen und vorhergesagten Instanzen kleiner, während die Anzahl der von der Hypothese nicht abgedeckten Beispiele größer wird. Die Anzahl der von der Konklusion abgedeckten Instanzen bleibt gleich, da sich die Konklusion nicht ändert. Dieses Verhalten wird, wie oben schon erwähnt, zur Einschränkung des Hypothesenraums verwendet. Die gelernte Regelmenge besteht aus den vollständig instanziierten Regelmodellen, die von RDT akzeptiert werden.

2.2.3 Das Rule Restructuring Tool (RRT)

MOBALs Rule Restructuring Tool (RRT) hat die Aufgabe, die Struktur einer gegebenen Regelmenge zu analysieren und gegebenenfalls zu verbessern, ohne die inferentielle Hülle (Coverage) zu verändern.

Für die Analyse der bestehenden Struktur gibt es eine Reihe von Qualitätsmerkmalen, bezüglich derer die Regelmenge überprüft werden kann. Dazu gehören sowohl semantische als auch syntaktische Kriterien wie z.B. die Anzahl der Regeln und die Anzahl der Literale als syntaktische Merkmale sowie z.B. die durchschnittliche Anzahl abgedeckter Instanzen als semantisches Kriterium.

Desweiteren werden Statistiken über die Häufigkeiten von Prädikaten, Fakten, Regeln und Sorten sowie Erklärungen zur Coverage bestimmter Fakten und Regeln angeboten. Zu einer Regel können beispielsweise alle Fakten aus der Wissensbasis angezeigt werden, die von ihr abgedeckt bzw. nicht abgedeckt werden. Für ein gegebenes Faktum kann auch untersucht werden, warum es von der entsprechenden Regel abgedeckt bzw. nicht abgedeckt wird. Dazu wird vom System dargestellt, welche Prämissen für das Beispiel erfüllt sind und welche nicht erfüllt werden können, so daß der Benutzer beispielsweise mögliche Fehler oder Inkonsistenzen in der Wissensbasis entdecken kann.

Im Anschluß an die Analyse der aktuellen Struktur gibt es drei Wege für eine Restrukturierung der Regelmenge, die durch das Tool RRT unterstützt werden:

- das Entfernen von redundanten Regeln (siehe auch 4.7),
- das „Glätten“ der Inferenzstruktur sowie
- das „Deepening“ der Inferenzstruktur.

Durch das „Glätten“ werden Konzepte durch ihre Definitionen ersetzt, während beim „Deepening“ Prämissen einer Regel durch ein Konzept zusammengefaßt werden. Es handelt sich damit also um zwei komplementäre Strukturierungsmaßnahmen. Zur Verdeutlichung betrachte man das folgende Beispiel:

```
r1: sterblich(X) & not(tier(X)) → mensch(X).
r2: mensch(X) & männlich(X) & vorfahre(X,Y) → vater(X,Y).
```

Eine „Glättung“ der Inferenzstruktur kann durch die Regel `r3` erfolgen:

```
r3: sterblich(X) & not(tier(X)) & männlich(X) & vorfahre(X,Y)
→ vater(X,Y).
```

> Das Vertiefen, also der umgekehrte Prozeß, erfolgt, indem man `r3` durch `r1` und `r2` ersetzt.

Das Vertiefen einer Inferenzstruktur führt also zu einer Modularisierung der Regelmenge. Für eine genauere Darstellung der Tools RRT, insbesondere der Restrukturierung der Regelmenge, siehe [Morik et al, 1993].

2.2.4 Der Modellierungs– und Lernzyklus

Dieser Abschnitt erläutert das Zusammenspiel von Modellierungs– und Lernphasen. Es ergibt sich ein zyklischer Ablauf dieser Phasen, der mit einem Edit–Compile–Debug Zyklus verglichen werden kann.
Der Modellierungsprozeß beginnt mit der Festlegung eines Rahmens für das Modell. Darin werden erste Beobachtungen in Form von Fakten umgesetzt. Durch das Hinzufügen weiterer Wissenselemente z.B. in Form von Regeln kann die Wissensbasis ergänzt und vervollständigt werden.
Das erstellte Modell ist zu jedem Zeitpunkt operational, d.h. das vorhandene Verhalten kann in Form von Experimenten überprüft werden. Ein solches Experiment kann beispielsweise darin bestehen, ein Faktum in das System einzugeben und die dadurch neu inferenzierten Fakten zu untersuchen.
Bevor die erste Lernphase beginnen kann, müssen Regelmodelle (siehe oben) in das System eingegeben und die Parameter, insbesondere das beschriebene Akzeptanzkriterium, vorgegeben werden, sofern nicht die Standardparametereinstellungen verwendet werden sollen.
Es ist üblich, die Wissensbasis in ein Lern– und ein Testset zu unterteilen. Mit dem Lernset (Trainingsmenge) werden die Lernläufe durchgeführt, während das Testset dazu dient, die gelernte Regelmenge zu validieren. Häufig sind Anwendungen dadurch gekennzeichnet, daß nur eine begrenzte, relativ kleine Anzahl von Beispielen vorliegt. Aus diesen Beispielen sollen dann Regeln gelernt werden, die später auftretende Situationen korrekt behandeln. Die Zerlegung in ein Test– und Lernset ist in diesen Fällen problematisch.
Eine Möglichkeit, die sich hier anbietet, ist das Ziehen einer Beispielmenge mittels einer Cross Validation (siehe [Breiman et al, 1984]). Auf diese Weise wird die Beispielmenge in eine bestimmte Anzahl (v) gleich großer Mengen unterteilt. Es werden dann v Lernläufe durchgeführt, wobei jeweils eine der Mengen als Testset und alle anderen als Lernset verwendet werden. Die gelernten Regeln müssen dann auf das Testset angewendet werden, wobei man die Ergebnisse anschließend mit der korrekten Klassifikation vergleicht.
Aber auch dieses Vorgehen kann sich bei sehr geringer Beispielanzahl als problematisch erweisen (siehe 4.9). Von besonderer Bedeutung ist in dieser Arbeit daher die Validierung der

Regeln durch die Experten bzw. durch die reale Anwendung der Regeln.
Für die Validierung der gelernten Regeln, die zur Lernphase gehört, müssen Kriterien aufgestellt werden, anhand derer die Qualität der Regeln bewertet werden kann. Ergibt diese Analyse, daß kein zufriedenstellendes Ergebnis erreicht werden konnte, müssen die vorher getroffenen Entscheidungen gegebenenfalls revidiert bzw. korrigiert werden. Einige der möglichen Ansatzpunkte für Revisionen sind:

1. Modellierungsentscheidungen, z.B.
   - Sind alle relevanten Elemente des Sachbereichs berücksichtigt worden?
   - Sind ggf. Kategorieneinteilungen ungünstig gewählt worden?
2. Repräsentationsentscheidungen z.B.
   - Welche Elemente werden als Prädikate repräsentiert, welche als Sorten?
3. Auswahl geeigneter Metaprädikate:
   - Wurde das gewünschte Lernergebnis verfehlt, weil die Wahl der Metaprädikate ungünstig war?
4. Wahl des Akzeptanzkriteriums:
   - Hätten mit einem strengeren / schwächeren Akzeptanzkriterium bessere Lernergebnisse erzielt werden können?

Revisionen können mit Unterstützung des Systems MOBAL durchgeführt werden. Eine wichtige Operation ist dabei z.B. das Löschen von Regeln. Hier kann es sich beispielsweise um Regeln handeln, die eine ungünstige Kategorieneinteilung bilden. Das System überprüft dann, welche Fakten durch diese Regeln abgeleitet worden sind, und entfernt diese aus der Wissensbasis. Umrepräsentationen können gegebenenfalls durch die Eingabe neuer Regeln erfolgen. Man betrachte dazu das folgende Beispiel:

> Es gibt in der Wissensbasis ein Prädikat mit der Deklaration:
>
> pred(<Sorte_1>,<Sorte_2>).
>
> Es hat sich nun herausgestellt, daß diese Repräsentation ungünstig ist und aus diesem Prädikat neue Prädikate entstehen sollen, in denen das zweite Argument wegfällt. Weiter sei angenommen, daß die Sorte zwei aus vier Argumenten $(c_1, \ldots, c_4)$ besteht, die nun in die Prädikatnamen einfließen sollen. Dies kann durch die Regeln

$$\text{pred(Y,X) \& eq}(X,c_1) \rightarrow c_1\text{_pred(Y)}$$

...

$$\text{pred(Y,X) \& eq}(X,c_4) \rightarrow c_4\text{_pred(Y)}$$

erreicht werden. Es gibt damit vier neue einstellige Prädikate mit den Namen c_1_pred bis c_4_pred.

Sind die nötigen Revisionen am operationalen Modell vorgenommen worden, so wird der Zyklus erneut durchlaufen. Der Prozeß kann solange fortgesetzt werden, bis das gewünschte Ergebnis erreicht werden konnte.

2.2.5 Diskussion

Der Benutzer von RDT hat die Möglichkeit, den Lernvorgang durch Erkenntnisse über semantische Strukturen und Zusammenhänge im Sachbereich zu steuern, indem er geeignete Metaprädikate vorgibt. Ist aber über den Sachbereich wenig bekannt, so ist die Hypothesenraumeinschränkung durch Regelmodelle kritisch. Es kann vorkommen, daß keine zufriedenstellenden Regeln gelernt werden können, da die „richtigen“ Metaprädikate nicht angegeben wurden. Eine gewisse Vorstellung von dem Lernziel sollte daher vorhanden sein, um die gewünschten Ergebnisse nicht zu verfehlen. In dem hier zugrundeliegenden Sachbereich stellte sich dieser Aspekt als völlig unproblematisch heraus (siehe Kapitel 4).
Wie bereits aus früheren Erfahrungen deutlich geworden ist, stellt die Größe der Wissensbasis ein erhebliches Problem für das System dar. Dies bezieht sich sowohl auf das Tool RDT als auch auf Operationen zur Modellierung der Wissensbasis (z.B. das Entfernen von Regeln).
Abgesehen von diesem, teilweise aber sehr gravierenden Problem wurde der Einsatz von MOBAL auch durch die Experten als sehr positiv bewertet. Durch die Verwendung der Tools konnte die Modellierung umfassend unterstützt werden. Die Repräsentation des operationalen Modells erwies sich als durchaus angemessen für die Kommunikation mit den Experten.

2.3 Delphi / ObjectVision

Zur Implementierung des wissensbasierten Systems wurde ein objektorientierter Ansatz gewählt, der auf das Softwarewerkzeug Delphi[4] (siehe [Borland, 1994]) zurückgreift. Delphi ist eine integrierte Entwicklungsumgebung (IDE), die die Konstruktion von Microsoft Windows–Anwendungen unterstützt. Delphi baut auf Komponenten auf, die in Klassenbibliotheken für den Benutzer zur Verfügung stehen.

[4] Delphi ist der Nachfolger des Tools ObjectVision (siehe [Borland, 1991]), mit dem eine erste Implementierung des Systems CORA vorgenommen wurde.

Abbildung 2.9: Die Benutzeroberfläche von Delphi

2.3.1 Formulare und Komponenten

Formulare stehen im Mittelpunkt der Anwendungen, die mit Delphi entwickelt werden. Eine Anwendung besteht aus einem Hauptformular und einer Reihe von untergeordneten Formularen. Ein Formular ist eine Komponente einer bestimmten Klasse (TForm), die wiederum andere Komponenten enthalten kann. Es kann wie eine Zeichenfläche verwendet werden, indem hierauf Komponenten plaziert und ausgerichtet werden, um einzelne Teile der Benutzerschnittstelle zu entwickeln. Das Hauptformular der Anwendung und dessen Komponenten kommunizieren mit anderen Formularen und deren Komponenten.

Die Visual Component Library (VCL) von Delphi enthält die zur Verfügung stehenden Komponenten. Dieser Library können eigene Komponenten hinzugefügt, oder es können Komponenten aus ihr gelöscht werden, indem die mit den Komponenten verbundenen Unit–Dateien hinzugefügt bzw. entfernt werden.

Die grafische Benutzeroberfläche von Delphi (siehe Abb. 2.9) zeigt eine Komponentenpalette an, die die zur Verfügung stehenden Komponenten darstellt. Komponenten schließen sowohl alle sichtbaren Bestandteile einer Anwendung ein, wie z.B. Dialogelemente und Schaltflächen, als auch die während der Programmausführung unsichtbaren Elemente wie z.B. System–Timer, die Zeitintervalle bereitstellen, die an Ereignisse angehängt werden können.

Aus der Komponentenbibliothek können komplette Dialogelemente ausgewählt werden, z.B. Dialoge zum Öffnen oder Speichern von Dateien. Weitere vordefinierte Dialoge sind die Font– und Farbauswahldialoge, die Dialogfenster zum Drucken und zum Einstellen des Druckers sowie Dialoge zum Suchen und Ersetzen. Diese Dialogelemente entsprechen den Windows–Standards.
Häufig verwendete Komponenten sind beispielsweise Felder, Menüs, Elemente zur Beschriftung, Tabellen und Elemente zur Gruppierung von Komponenten und Grafiken. Es können eine Reihe von Feldarten unterschieden werden. „Editier"– und „Memofelder" erlauben die Anzeige eines ein– bzw. mehrzeiligen Bereichs, in dem Textzeilen angezeigt und vom Anwender eingegeben oder geändert werden können. Weiter gibt es Felder und Listen, die auszuwählende Optionen für den Anwender in unterschiedlicher Form bereitstellen. Gruppierungselemente werden häufig zur Erstellung von Symbolleisten und Statuszeilen verwendet, indem z.B. Schalter zu einer Schalterleiste zusammengefaßt werden. Zu den neueren Windows–Elementen gehören die „Arbeitsblattregister", die es ermöglichen, einem Formular das Aussehen von Seiten zu geben. Durch die „Seitenregister" am oberen oder unteren Rand können die Seiten ausgewählt werden. Als Schaltflächen gibt es einfache Schalter, Schalter, die mit Symbolen und Text versehen werden und kleinere Schalter mit Symbolen, die zur Konstruktion einer Systemleiste verwendet werden können.

2.3.2 Eigenschaften von Komponenten

Alle Komponenten von Delphi sind Objekte, die sich durch eine Reihe von Eigenschaften auszeichnen. Der sogenannte „Objektinspektor" von Delphi (siehe Abb. 2.9 links) ermöglicht das Anpassen der Darstellung und des Verhaltens der Komponenten der Anwendung. Die Eigenschaften der im Formular gewählten Komponente werden im Objektinspektor angezeigt und können während der Programmentwicklung als Initialeigenschaften eingestellt werden. Diese Eigenschaften können während der Programmausführung durch den Quelltext geändert werden.
Im weiteren werden einige Eigenschaften vorgestellt, die für viele Komponenten anzugeben sind. Die wichtigste Eigenschaft ist der Name einer Komponente, der sie für das zugrundeliegende Programm identifizierbar macht. Davon zu unterscheiden ist die Eigenschaft „Caption", die die Beschriftung der Komponente enthält. Hier dürfen beliebige Zeichenketten (inkl. Leerzeichen) verwendet werden.
Eigenschaften, die das Aussehen der Komponenten beeinflussen sind beispielsweise „Color", „Font", „Borderstyle" und ähnliches. Die Größe und Positionierung der Komponenten wird durch die Eigenschaften „Width" und „Height" bzw. „Left" und „Top" bestimmt.
Die Eigenschaft „Enabled", die für alle Dialogelemente zur Verfügung steht, steuert die Reak-

tion der Komponente auf Maus–, Tastatur– und Timer–Ereignisse. Das Setzen von „Visible" macht die Komponente zur Laufzeit sichtbar bzw. unsichtbar.
Zur Konstruktion von kontext–sensitiver Hilfe kann für alle Elemente ein „HelpContext" angegeben werden. Bei Aktivierung der Hilfe wird dann das durch den Kontext angegebene Hilfe–Thema angezeigt.

2.3.3 Ereignisbehandlungen

Im Anschluß an die Erzeugung von Formularen mit passenden Komponenten werden Ereignisbehandlungsroutinen für die Komponenten erzeugt, so daß die Komponenten auf Ereignisse reagieren können, die während der Ausführung des Programms auftreten. Die verwendete Sprache für solche Routinen ist Object Pascal. Object Pascal basiert auf der Sprache Turbo Pascal, Version 7.0. Es sind lediglich einige Sprachkonstrukte hinzugefügt worden, siehe [Borland, 1994].
Mit der Seite „Ereignisse" des „Objektinspektors" können Routinen erzeugt werden. Diese werden aufgerufen, sobald ein Ereignis eintritt. Ereignisse sind Benutzeraktionen oder Ereignisse im System, die die Anwendung erkennen kann. Alle Komponenten haben verschiedene Ereignisse, auf die sie reagieren können, wie beispielsweise ein Mausklick auf die Komponente oder das Aktivieren, Schließen, Doppelklicken und Erstellen der Komponente.
Eine Ereignisbehandlung ist der Quelltext, der festlegt, wie eine Komponente auf ein Ereignis reagieren soll. Die Routinen rufen Funktionen, Prozeduren und Methoden, also Funktionen und Prozeduren, die mit einer Komponente verbunden sind, auf. Für Formulare gibt es beispielsweise die Methoden „Create", „Show" und „Close". Diese Methoden können von allen Formularen aufgerufen werden, die daraufhin erstellt, angezeigt bzw. geschlossen werden.
Bei der Erzeugung einer Ereignisbehandlung mit dem Objektinspektor werden Teile des Quelltextes von Delphi erzeugt und verwaltet. Der folgende Quelltext wird automatisch erzeugt, wenn der Schalter „Button1" in das Formular „Form1" eingefügt wird:

```
procedure TForm1.Button1Click(Sender: TObjekt);
begin

end;
```

Mit dem Quelltexteditor hat der Anwender Zugriff auf den gesamten Programm–Code in einem Anwendungsprojekt. Die Ereignisbehandlungsroutinen können durch den Aufruf von Methoden, Prozeduren und Funktionen vom Benutzer geschrieben werden:

```
procedure TForm1.Button1Click(Sender: TObjekt);
```

```
begin
  Form1.Close;
end;
```

Die hier definierte Prozedur ruft die Methode „Close“ des aufrufenden Formulars auf, d.h. das Formular wird nach dem Anklicken des Schalters geschlossen.

2.3.4 Entwicklungswerkzeuge

Delphi bietet eine Reihe von Entwicklungswerkzeugen, die den Benutzer bei der Erstellung einer Anwendung unterstützen, wie z.B. Projekt– und Formularschablonen. Projektschablonen bieten verschiedene Anwendungsentwürfe, die als Anfang beim Erstellen der Anwendungen verwendet werden können.

Dabei handelt es sich um vordefinierte Formulare, die bereits Komponenten und zum Teil zugehörige Ereignisbehandlungen (siehe 2.3.3) enthalten. Die Schablonen müssen lediglich an die Anwendung angepaßt werden.

In der Galerie stehen Formulare für einen Standarddialog, einen Auswahl– und einen mehrseitigen Dialog sowie eine Schablone für ein Infofenster bereit. Während das Standarddialogfenster nur die drei Schalter „OK“, „Abbruch“ und „Hilfe“ enthält, gibt es im Auswahldialog zusätzlich zwei Listen, deren Elemente mithilfe der vier Schalter [$>$], [$\gg$], [$\ll$] und [$<$] verschoben werden können. Die links stehende Liste fungiert dabei als Ausgangsliste, die rechte ist die Zielliste. Der mehrseitige Dialog enthält eine Registerkomponente, die die Darstellung mehrerer Formularseiten ermöglicht. Das Infofenster besteht aus einem Grafikfeld, Beschriftungselementen und einem „OK“–Schalter.

Eigene Projekte können auch als Projektschablonen abgespeichert werden, während Formularschablonen eine Auswahl an verschiedenen, vordefinierten Formularen zur Entwicklung der Oberfläche bieten. Selbst erstellte Formulare können auch als Formularschablonen abgespeichert werden.

Weitere Tools von Delphi sind u.a. die Projektverwaltung, ein Menü–Designer, ein integrierter Debugger, ein Objekt–Browser und ein Bild–Editor, die hier nicht weiter beschrieben werden. Für die Erstellung von Datenbankanwendungen stehen weitere Tools zur Verfügung.

2.3.5 Projekte

Ein vollständiges Projekt (eine Anwendung) besteht aus allen Dateien, die zur Erstellung der Zielanwendung benötigt werden:

- Projektdatei (.DPR)

- Unit–Datei (.PAS)
- Formulardatei (.DFM)
- Quelltextdatei für Units ohne Formular (.PAS).

Diese Dateien werden im folgenden näher erläutert.

Zu jedem Projekt existiert eine Projektdatei. In dieser Datei werden Informationen zu den Unit– und Formulardateien des Anwendungsprojekts geführt. Sie wird zu Beginn der Programmentwicklung durch Delphi erstellt und während der Entwicklung des gesamten Projekts verwaltet. Die Datei enthält den Projektnamen, eine Uses–Anweisung, in der die Units des Projekts und deren zugeordnete Formulare aufgelistet sind und einen Programmblock, der den Quelltext zur Ausführung der Anwendung und Aktivierung des Hauptformulars enthält.

Für jedes Formular wird eine eigene Unit–Datei und eine Formulardatei erzeugt. Die Unit–Datei enthält Object Pascal Quelltext, also die Ereignisbehandlungen, die die Funktionalität der Komponenten bestimmen. Zu Beginn besteht diese Datei aus dem Interface– und dem Implementation–Abschnitt und dem Programmblock mit dem Quelltext für die Initialisierung. Durch das Einfügen von Komponenten mit ihren zugehörigen Ereignisbehandlungsroutinen wird der Quelltext teilweise automatisch von Delphi, aber auch manuell durch den Benutzer erweitert.

Bei der Formular–Datei handelt es sich um eine Binärdatei, die nur in der Formularansicht durch die Visualisierung des Formulars und seiner Komponenten sichtbar gemacht werden kann. Es ist keine direkte Änderung am Quelltext dieser Datei durch den Benutzer vorgesehen.

Mithilfe von Units, die nicht zu einem Formular gehören, können beispielsweise Bibliotheken mit mathematischen Funktionen importiert oder separat compilierbare Units nicht visueller Objekte erstellt werden.

2.3.6 Entwicklung einer Windows–Hilfe

Ein Hilfesystem stellt dem Benutzer Online–Informationen über eine Anwendung zur Verfügung. Es kann aus der Applikation heraus gestartet werden und erscheint somit für den Benutzer als Teil der Anwendung. Für den Entwickler ist das Hilfesystem dagegen eine selbständige Windows–Anwendung, die wie jedes andere Programm unabhängig von anderen benutzt werden kann.

Das Hilfesystem besteht aus Text und Grafiken, die in Form eines Hilfefensters über die Anwendung gelegt werden können. Die Hilfethemen sind als Hypertext miteinander verknüpft. Sie enthalten Kreuzreferenzen zu verwandten Themen, so daß durch Anklicken eines Sprungbegriffs in Form eines hervorgehobenen Textstücks (Hotspot) die Beschreibung dieses Themas

aufgerufen wird. Diese Sprünge können ein neues Hilfethema aufrufen oder lediglich ein Pop–up–Window anzeigen, das eine kurze Beschreibung eines Begriffs oder Befehls enthält.

Die Hilfedatei wird gebildet aus Themendateien (darunter eine Themendatei, die ein Inhaltsverzeichnis enthält), Grafikdateien und einer Hilfeprojektdatei. Eine Themendatei besteht aus den formatierten Texten für die Hilfethemen und speziellen Steuercodes und Fußnoten. Die Grafikdateien enthalten Bitmaps und Metafiles, die in den Hilfethemen angezeigt werden können. Die Hilfeprojektdatei enthält alle Informationen, die der Microsoft Windows Help Compiler für die Umwandlung der Themen– und Grafikdateien in eine binäre Hilferessourcendatei benötigt. Die Ressourcendatei (mit der Endung .hlp) kann mithilfe des Programms „Windows Help“ geöffnet und angezeigt werden.

Im weiteren werden die Themendateien und die Projektdatei näher beschrieben. Ist die Erstellung des Hilfesystems abgeschlossen, muß die Einbettung in die Anwendung erfolgen, so daß die Hilfe von hier aus gestartet werden kann. Das Einbinden eines Hilfesystems in eine Delphi–Anwendung wird daher im Anschluß erläutert.

Themendateien

Jede Themendatei enthält ein oder mehrere Themen (Topics). Ein Topic ist eine Einheit von Informationen, die durch das Programm „Windows Help“ angezeigt wird.

Eine Themendatei wird erstellt, indem mithilfe eines Texteditors die notwendigen Steuercodes eingefügt werden oder einfacher durch eine Textverarbeitung, die RTF (Rich Text Format)–Dateien erzeugen kann (z.B. Word). Diese zweite Möglichkeit soll kurz vorgestellt werden.

Jedes Thema wird durch Einfügen eines manuellen Seitenumbruchs am Ende des Themas als eine Seite formatiert. Der Text kann mit den üblichen Mitteln gestaltet werden (Fonts, Farben, Linien etc.).

Wichtig sind die Fußnoten für jedes Hilfethema, durch die die Topics identifiziert werden können und die Navigation durch das Hilfesystem gesteuert wird. Die Funktionen der wichtigsten Fußnoten sind in der Tabelle 2.3 dargestellt.

Zur Definition von „Hotspots“, also Sprüngen zu verwandten Themen, muß der Sprungbegriff doppelt unterstrichen werden. Direkt nach dem Sprungbegriff, der in der Hilfe später grün und einfach unterstrichen erscheint, wird der „Kontext String“ des Zieltopics angegeben. Der Kontext String muß als versteckter Text formatiert werden.

Für das zu erstellende Inhaltsverzeichnis ist es sinnvoll, alle Titel der Topics des Hilfesystems als Hotspot aufzulisten, so daß von dieser Seite aus jedes Hilfethema erreicht werden kann.

Bezeichnung der Fußnote	Identifizierung des ...	Funktion
#	Kontext String	Eindeutige Identifizierung des Topics zur Spezifikation der Sprünge
$	Titel	Stellt den Titel des Topics in der Dialogbox Suchen und in der History–Liste dar
K	Schlüsselworte	Schlüsselworte, die in der Dialogbox Suchen aufgeführt werden (oberes Fenster)
+	Browse–Sequenz	Bestimmt die Reihenfolge der Topics, wenn der Benutzer die ≪– und ≫–Schalter benutzt.

Tabelle 2.3: Fußnoten und ihre Funktionen

Hilfeprojektdatei

Die Hilfeprojektdatei ist ein Textfile (mit den Endung .hpj), das u.a. folgende Bestandteile umfaßt:

FILES: Dieser Abschnitt ist der einzige obligatorische Teil der Projektdatei. Er legt fest, welche Themendateien zu dem Hilfesystem gehören sollen.

OPTIONS: In diesem Abschnitt können einige Optionen angegeben werden, z.B.

- **CONTENTS:** Der Kontext String des Inhaltsverzeichnisses, also das Thema, das durch das Anklicken des Schalters Inhalt aufgerufen wird, kann hiermit bestimmt werden.
- **TITLE:** Ein Titel für das Hilfesystem kann angegeben werden.
- **COMPRESS:** Mit dieser Option wird der Kompressionsgrad der compilierten Hilfedatei bestimmt.

MAP: Hier können die Kontext Strings mit einer Kontextnummer verknüpft werden.

BITMAP: Die zu dem Hilfesystem gehörenden Grafikdateien können in diesem Anschnitt angegeben werden.

Einbinden eines Hilfesystems in eine Delphi–Anwendung

Über die Eigenschaft „HelpFile“ des Objekts Application, also der Anwendung, kann eine Hilfedatei (.hlp) für die Anwendung angegeben werden. Der Aufruf des Hilfesystems muß nicht über die recht umständliche Funktion „WinHelp“ (aus Windows API) erfolgen. Einfacher ist die Verwendung der Methoden „HelpJump“ und „HelpContext“, die von Delphi bereitgestellt werden.

Die Methode „HelpJump (**const** JumpID : **string**) : Boolean“ erhält als Parameter einen Kontext String und zeigt den mit diesem „Kontext String“ verbundenen Hilfebildschirm aus der Hilfedatei an. Ist der Eigenschaft „HelpFile“ zuvor keine Hilfedatei zugewiesen worden, so gibt „HelpJump“ False zurück, sonst True.

Ähnlich arbeitet auch die Methode „HelpContext(Context : THelpContext) : Boolean“, die aber als Parameter die Kontextnummer erhält. Eine Kontextnummer kann durch die Eigenschaft „HelpContext“ fast allen Komponenten der Delphi–Anwendung zugewiesen werden.

Ist in der Anwendung eine Komponente selektiert, so wird nach Betätigen von [F1] ein der Kontextnummer entsprechender Hilfebildschirm angezeigt. Auf diese Weise können kontext–sensitive Hilfen realisiert werden.

2.3.7 Erfahrungen mit Delphi

Wie bereits zu Anfang des Abschnitts erwähnt wurde, ist die Implementierung zunächst mit dem Tool ObjectVision erfolgt. Aufgrund einiger Schwächen dieses Systems wurde dann auf das System Delphi zurückgegriffen.

Beiden Tools ist gemeinsam, daß sie die schnelle Entwicklung eines Prototypen für das WBS sehr gut unterstützen können. Die Gestaltung einer Oberfläche ist damit völlig problemlos, und erste Ergebnisse sind sofort sichtbar. Für die beiden wesentlichen Aspekte in der Konstruktion dieses WBS, nämlich für die Modellierung der Expertise und für die Gestaltung einer benutzerfreundlichen Oberfläche, existiert daher jeweils eine Grundlage zur Kommunikation mit den Experten in Form eines operationalen Modells (in MOBAL) bzw. eines Prototypen (implementiert mit Delphi). Dieser Aspekt ist für beide Seiten sehr motivationsfördernd und hat wesentlich zum Gelingen der Arbeit beigetragen.

Mit dem Tool ObjectVision haben sich dennoch einige Probleme ergeben, die nun kurz dargestellt werden. Durch die immer größer werdende Komplexität des Systems CORA stellte es sich als extrem nachteilig heraus, daß ObjectVision über keinen integrierten Debugger verfügt. Die Fehlersuche gestaltete sich daher oft außerst problematisch.

Das Konzept, über Dynamic Link Libraries (DLLs) auf eigene Funktionen zuzugreifen, die in Turbo Pascal bzw. C++ geschrieben wurden, erhöhte die Flexibilität des Systems. Es gab

aber häufig Schwierigkeiten mit der Übergabe dieser Funktionen, die wegen der mangelnden Transparenz des Systems teilweise schwer zu lösen waren.

Diese Probleme konnten sämtlich durch den Einsatz des Systems Delphi gelöst werden. Die Struktur des Tools ist für den Anwender wesentlich transparenter. Der direkte Zugriff (von Delphi) auf Turbo Pascal erwies sich als flexibler im Vergleich zum DLL–Konzept.

Außerdem ergaben sich durch eine erweiterte Komponentenbibliothek einige neue Gestaltungsmöglichkeiten für die Benutzeroberfläche, die mit ObjectVision nicht oder nur in sehr aufwendiger Weise möglich gewesen sind.

Als bislang einziger Schwachpunkt des Tools Delphi (in der Version 1.0) kann hier das noch nicht ganz fehlerfreie Hilfesystem und die dürftige Dokumentation festgehalten werden. Grenzen für die Anwendbarkeit des Systems wurden in diesem Rahmen noch nicht erreicht, im Gegenteil: es bleiben hier viele Features des Systems Delphi, wie beispielsweise das Datenbank–Tool, ungenutzt.

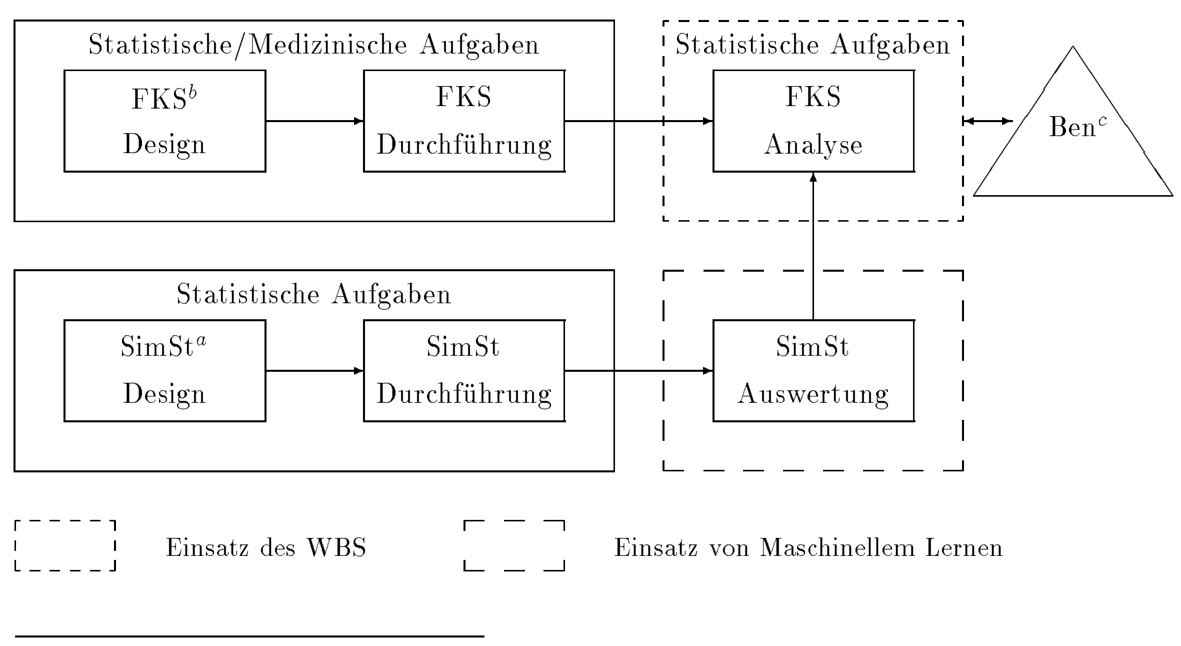
Kapitel 3

Knowledge Level Modelle des wissensbasierten Systems CORA

KADS, die in Abschnitt 2.1 vorgestellte Methodik zur Konstruktion wissensbasierter Systeme, wird in dem folgenden Kapitel zur Spezifikation des Systems CORA verwendet. Der KADS–Ansatz faßt die Wissensakquisition als einen Modellierungsprozeß auf, der die Erstellung einer Reihe von Modellen einschließt. Dieses Kapitel besteht aus der Beschreibung der von KADS geforderten Knowledge Level Modelle für das System CORA, die in Abschnitt 2.1.3 eingeführt worden sind. Die Übertragung dieser Modelle auf eine symbolische Ebene wird in Kapitel 5 erläutert.

3.1 Organisationsmodell

Der folgende Abschnitt liefert eine Beschreibung des Umfelds, in dem das wissensbasierte System CORA eingesetzt werden soll.
Wie bereits in der Einleitung kurz erwähnt, soll das System die Analyse von Fall–Kontroll–Studien unterstützen. Diese Studien werden vorwiegend im Bereich der Epidemiologie, also der Gesundheitsforschung, entworfen und durchgeführt. Sie dienen u.a. zur qualitativen sowie quantitativen Erfassung von Risikofaktoren für Gesundheitsschäden großer Bevölkerungsgruppen. Da sowohl solche Risiken als auch die Gesundheitsschäden häufig in Kategorien erfaßt werden, können die Beobachtungen zur weiteren Analyse in sogenannten Kontingenztafeln zusammengefaßt werden.
Im Rahmen einer Auswertung solcher Kontingenztafeln stehen verschiedene konkurrierende statistische Methoden u.a. zur Abschätzung des Risikos einer Erkrankung zur Verfügung. Von diesen Verfahren sind in der Regel die theoretischen statistischen Eigenschaften bekannt. Da es sich dabei aber meist um asymptotische Resultate handelt, werden typischerweise Si-



[a]Simulationsstudie
[b]Fall–Kontroll–Studie
[c]Benutzer

Abbildung 3.1: Organisationsmodell

mulationsstudien entworfen und durchgeführt, um einen Eindruck über das finite Verhalten der Verfahren zu erhalten. In dieser Arbeit wird eine Auswertung solcher Studien mithilfe eines wissensbasierten Ansatzes vorgestellt, der auf Methoden der Künstlichen Intelligenz basiert. Mithilfe des maschinellen Lernens wird eine Charakterisierung von einigen in der Kontingenztafelanalyse zu verwendenden statistischen Methoden gewonnen, die von den Dateneigenschaften der Fall–Kontroll–Studie auf die Eignung der Methoden schließt.
In der Organisation sind eine Reihe von Aufgabenbereichen zu unterscheiden (siehe auch Abb. 3.1). In das direkte Umfeld des Systems CORA gehören:

- das Design und die Durchführung der Datenerhebung für die Fall–Kontroll–Studien,
- die Analyse von Fall–Kontroll–Studien mittels einer Kontingenztafelanalyse,
- das Design und die Durchführung von Simulationsstudien und
- die Auswertung von Simulationsstudien.

Außerdem sind in der Organisation noch zwei weitere Aufgabenbereiche zu finden, nämlich

- die Herleitung der statistischen Verfahren und

- die Untersuchung ihrer theoretischen Eigenschaften.

Diese beiden Bereiche sind aber für das wissensbasierte System lediglich von geringem Interesse und werden daher im weiteren nur am Rande behandelt. Eine detaillierte Beschreibung des Sachbereichswissens aus den vier Hauptaufgabenbereichen erfolgt im Abschnitt 3.3.1.
In der Organisation des Systemumfelds ist die Interdisziplinarität der Aufgabenbereiche von zentraler Bedeutung. Eine Reihe von Aufgaben erfordert die Kooperation von Experten bzw. die Vereinigung von unterschiedlichen Wissensbereichen in einem einzelnen Experten. Das Design der Fall–Kontroll–Studie, also die Planung der Datenerhebung, verlangt sowohl Kenntnisse des medizinischen Hintergrunds als auch statistisches Wissen. Speziell für die Datenanalyse sind jedoch sehr umfangreiche statistische Kenntnisse erforderlich, die ebenfalls, d.h. neben dem fachspezifischen Wissen, bei der Interpretation der gewonnenen statistischen Ergebnisse eine nicht unbedeutende Rolle spielen. Da Epidemiologinnen in der Regel hauptsächlich eine medizinische Ausbildung haben, ist eine Kooperation mit Statistikerinnen wünschenwert.
Im Idealfall arbeiten Statistikerinnen und Medizinerinnen von Beginn an zusammen. Die Statistikerin entwirft unter Berücksichtigung der Fragestellung und der äußeren Umstände ein Studiendesign, auf dessen Grundlage die Medizinerin die Datenerhebung durchführt. Die gesammelten Daten werden dann von der Statistikerin geeignet analysiert. Entscheidet man sich hier für die Durchführung einer Kontingenztafelanalyse, so wird die Statistikerin gegebenenfalls die in der Literatur beschriebenen Ergebnisse von Simulationsstudien untersuchen bzw. sogar selbst eine Simulationsstudie durchführen, sofern sie nicht über genügend Erfahrung in der Analyse von Kontingenztafeln verfügt. Werden die Daten der Fall–Kontroll–Studie auf diese Art erhoben und analysiert, handelt es sich um eine komplexe Aufgabe, die vielschichtige Kenntnisse erfordert.
In der Praxis ist aber häufig zu beobachten, daß Statistikerinnen erst nach abgeschlossener Datenerhebung hinzugezogen werden. Im Extremfall wird auf die Unterstützung durch Statistikerinnen völlig verzichtet. Dann hängt es von der Qualifikation der Medizinerinnen ab, ob die Analyse unter statistischen Gesichtspunkten korrekt und in diesem Sinne auch erfolgreich durchgeführt werden kann. Durch eine epidemiologische bzw. biometrische Ausbildung erhalten Medizinerinnen zusätzlich statistisches Wissen. Aber dies ist i.a. nicht so umfangreich, als daß die Unterstützung durch eine Statistikerin überflüssig würde. Bei mangelnder statistischer Qualifikation der durchführenden Personen können sich einige Probleme ergeben, die häufig in der Wahl einer, der gegebenen Datensituation nicht angemessenen, statistischen Methode resultieren, deren Anwendung dann zu verfälschten Ergebnissen führen kann. Eine Nutzung solcher Ergebnisse für weitere Entscheidungen kann fatale Folgen haben.
Die Datenanalyse wird, unabhängig von der durchführenden Person, in der Regel rechner-

gestützt vorgenommen. Die Standard–Software für statistische Auswertungen ist das SAS–System. SAS ist ein flexibles und daher sehr komplexes Programm–Paket, mit dem u.a. auch Fall–Kontroll–Studien analysiert werden können. Die Benutzung des Systems erfordert erhebliche Einarbeitung in die Progammierung von SAS und Statistik–Kenntnisse, hier also Erfahrungen mit dem Ablauf einer Kontingenztafelanalyse. Zielgruppe für dieses System sollten deshalb vor allem Statistikerinnen sein, da bei mangelnder statistischer Qualifikation fehlerhafte Analysen nicht ausgeschlossen sind.

Ein Beispiel für Systeme, die eine Programmierung durch eine grafische Benutzeroberfläche ersetzen, ist das Programm Statistica. Auch hier handelt es sich um ein sehr komplexes Programmpaket, mit dem eine Reihe von statistischen Auswertungen durchgeführt werden können. Ein sich daraus ergebender Vorteil des Systems ist die große Flexibilität. Nachteilig ist aber, daß auch hier relativ hohe Anforderungen an die Statistik–Kenntnisse der Benutzer gestellt werden. Aber gerade durch die einfache Bedienung des Systems wird es häufig von Anwendern eingesetzt, die nur gelegentlich eine statistische Analyse durchführen und daher über wenig diesbezügliche Erfahrungen verfügen.

Durch die Konstruktion eines wissensbasierten Systems wird versucht, in diese komplexe Organisation unterstützend einzugreifen. Die Auswirkungen, die die Einführung des Systems auf diese Organisation voraussichtlich haben wird, werden im Anwendungsmodell (siehe 3.2) beschrieben.

3.2 Anwendungs– und Aufgabenmodell

Der folgende Abschnitt beschreibt die Anforderungen an das System bezüglich seiner Funktionalität. Es wird angedeutet, welche Aufgaben durchzuführen sind, um diese Funktionalität zu erreichen. Das Aufgabenmodell wird damit in diesem Abschnitt nur sehr grob umrissen: die weitere Zerlegung der Unteraufgaben sowie die Zuweisung dieser Aufgaben an die Agenten wird erst nach der Einführung der wesentlichen Konzepte und Begriffe des Sachbereichs, im Aufgabenwissen des Expertisenmodells (siehe Abschnitt 3.3.3) detailliert beschrieben. Darüber hinaus werden in diesem Abschnitt externe Forderungen wie beispielsweise Hard– und Softwarevoraussetzungen festgehalten. Abschließend wird in einer Art Technology Assessment der Einfluß, den die Konstruktion des Systems haben wird, vorausgesagt und bewertet.

Das zu erstellende wissensbasierte System soll den im Organisationsmodell erwähnten Aufgabenbereich *Analyse von Fall–Kontroll–Studien* unterstützen, genauer den gesamten Prozeß einer Kontingenztafelanalyse bestehend aus der Schichtung der Daten, der Homogenitätsanalyse, der Schätzung des gemeinsamen Odds Ratios und der Durchführung von Unabhängig-

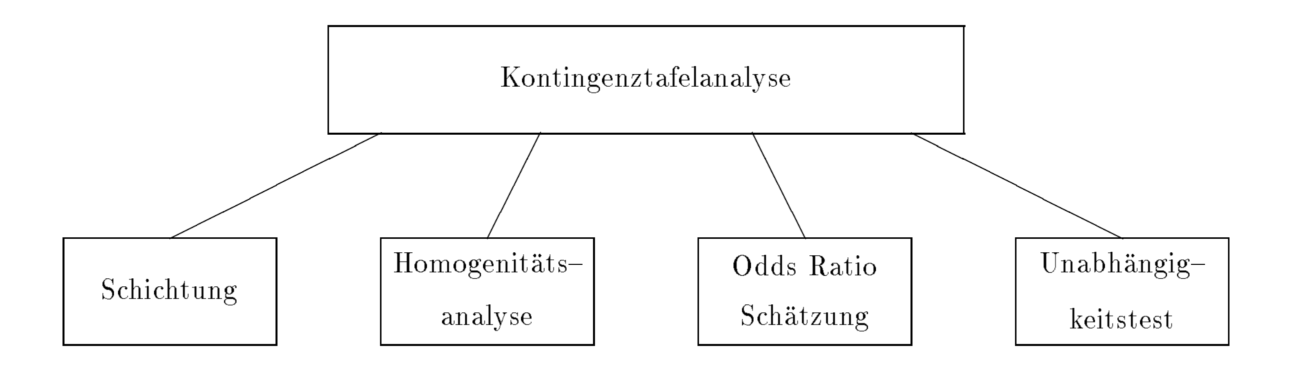


Abbildung 3.2: Aufgabenzerlegung für die Kontingenztafelanalyse

keitstests. Damit ergibt sich die in Abb. 3.2 dargestellte Zerlegung der Aufgaben für die Kontingenztafelanalyse. Der genaue Ablauf dieses Prozesses wird im Rahmen des Sachbereichs– bzw. des Aufgabenwissens in den Abschnitten 3.3.1 und 3.3.3 dargetellt. Alle Verfahren, die zur Analyse des Datensatzes notwendig sind, werden innerhalb des Systems implementiert und sollen von einer einheitlichen grafischen und benutzerfreudlich gestalteten Oberfläche aufrufbar sein. Es wird auf keine vorhandene statistische Auswertungssoftware zurückgegriffen.

Wichtig ist, daß dem Anwender ein Hilfesystem zur Verfügung steht, das einerseits die Benutzung des Programms erklären soll, andererseits auch die „inhaltlichen“ (statistischen) Aspekte ausführlich darstellt.

Das Design von Fall–Kontroll–Studien und die Datenerhebung werden nicht unterstützt. Voraussetzung zur Anwendung des Systems ist also das Vorliegen eines geeigneten Datensatzes. Aus diesem Datensatz kann mithilfe des Systems CORA ein Datensatz für eine Pilotstudie erstellt werden. Anhand der Daten der Pilotstudie wird eine erste Analyse der Daten durchgeführt. Diese Analyse läßt Schlüsse über die Dateneigenschaften zu, die zur Auswahl einer geeigneten Regel, d.h. zur Empfehlung einer statistischen Methode herangezogen werden können. Da die Daten der Pilotstudie dann nicht zur eigentlichen Datenauswertung verwendet werden dürfen, wird zur Erstellung der Originaldatensatz in zwei Datensätze unterteilt: die Pilotdaten und die Analysedaten. Als Methoden für die Aufteilung werden vom System eine einfache und eine geschichtete Zufallsauswahl zur Verfügung gestellt. Der Analysedatensatz wird für die Durchführung der Kontingenztafelanalyse verwendet; der Pilotdatensatz dient zusammen mit dem Analysedatensatz als Grundlage für eine im folgenden genauer vorgestellte Beratung der Anwender.

Die im Organisationsmodell erwähnten Aufgabenbereiche, die sich mit den Simulationsstudien beschäftigen, dienen zur Entwicklung der erwähnten Beratung, die im Rahmen des Systems

vom Anwender aufgerufen werden kann. Durch die Beratung wird der Benutzer des wissensbasierten Systems bei der Auswahl einer geeigneten Methode zur Risikoschätzung im Rahmen der Kontingenztafelanalyse unterstützt. Sie basiert auf einer Charakterisierung der Schätzer, die in Form einer Regelmenge durch die Auswertung der Simulationsstudien vorliegt. Die Regeln schließen von den Eigenschaften der Daten, also den Daten der Fall–Kontroll–Studie, auf eine Eignung der Schätzer. Zur Erzeugung einer Empfehlung wird u.a. anhand der Pilotstudiendaten überprüft, ob Regeln existieren, die eine Aussage über die vorliegende Datensituation machen. Gibt es entsprechende Regeln, so kann dem Benutzer die aus diesen Regeln abgeleitete Eignung der Schätzer mitgeteilt werden.

Die Benutzerfreudlichkeit des Systems zeichnet sich damit also durch drei Aspekte aus, nämlich durch die benutzerfreundliche Oberfläche, das Hilfesystem und zusätzlich durch eine Beratungskomponente zur Empfehlung geeigneter Methoden. Zur Umsetzung des dritten Aspekts wurde es als sinnvoll erachtet, auf Techniken zur Konstruktion wissensbasierter Systeme zurückzugreifen. Damit kann die Wissenserhebung, die Modellierung und die flexible Einkodierung des benötigten statistischen Wissens in Form einer lokalen Regelmenge erfolgen. Somit werden unterschiedliche Methoden zur Umsetzung des Systemkonzepts verbunden, um das gewünschte Ziel zu erreichen.

Im Hinblick auf eine Minimierung der Anforderungen an die Benutzung des Systems CORA scheint eine Anwendung, die auf PC's unter Microsoft Windows läuft, optimal.

Besondere Bedeutung hat auch die Einbeziehung des Zielgruppenprofils. Vom Benutzer des Systems CORA werden weder Kenntnisse in Bezug auf die Programmbenutzung noch umfassende Statistik–Kenntnisse verlangt.

Im Organisationsmodell wurde bereits erwähnt, daß eine Kooperation von Medizinerinnen und Statistikerinnen in der Praxis nicht immer stattfindet. Durch die Einführung des Systems wird diese Problematik nicht gelöst: die Medizinnerinnen, die bislang auf die Unterstützung durch Statistikerinnen verzichtet haben, werden ihr Verhalten wahrscheinlich nicht ändern.

Führen sie die Analyse ihrer Daten aber mithilfe des Systems CORA durch statt beispielsweise mit Standard Softwarepaketen der Statistik, so können sie von dem im System vorhandenen Know–How profitieren und die Ergebnisse ihrer Analyse dadurch u.U. verbessern. Vorteilhaft ist auch, daß sie sich nicht in ein komplexes System einarbeiten müssen.

Problematisch bleibt aber weiterhin, daß schon die Design–Phase, die vom System nicht unterstützt wird, fehlerhaft durchgeführt worden sein kann, wenn auf die Unterstützung einer Statistikerin verzichtet wird. Diese Fehler können auch durch eine korrekte Analyse, im Sinne von Garbage In – Gabage Out, nicht mehr behoben werden. Im Idealfall ist auch nach der Einführung des Systems eine Kooperation von Medizinerinnen und Statistikerinnen schon deswegen zu empfehlen, damit solche Designfehler vermieden werden können.

Wirkliche Änderungen können sich also nur in der Analysephase ergeben. Waren vorher Medizinerinnen mit geringen Statistik–Kenntnissen hier in jedem Fall auf statistische Hilfe angewiesen, so wird nun eventuell eine Verschiebung der Aufgabenverteilung zu beobachten sein. Nachdem die Statistikerin zur Durchführung einer Kontingenztafelanalyse geraten hat, kann eine Epidemiologin die Analyse gegebenenfalls selber ausführen, wobei weiterhin zumindest zur Interpretation der Ergebnisse ein Dialog mit Statistikerinnen mehr als wünschenswert wäre.

Es ist aber durchaus auch denkbar, daß die Analyse weiterhin von Statistikerinnen durchgeführt wird. Für Statistikerinnen, die über wenig Erfahrung in der Kontingenztafelanalyse verfügen, kann auch hier der Einsatz des Systems ratsam sein. Es ist aber zu beachten, daß die Flexibilität durch das System eingeschränkt werden kann: es wird lediglich eine bestimmte Auswahl an Methoden zur Verfügung gestellt. Nur wenn die Statistikerin diese Auswahl akzeptiert, wird sie das System anwenden. Die mit der Kontingenztafelanalyse häufig beschäftigten Statistikerinnen werden vermutlich auf ihre eigene Methodenbibliothek zurückgreifen und die eigenen Erfahrungen, die sie mit der Auswahl geeigneter Methoden gemacht haben, nutzen wollen. In solchen Fällen stellt das System jedoch eine sinnvolle Ergänzung dar.

3.3 Expertisenmodell (Model of Expertise)

In den folgenden Abschnitten wird die Problemlösungsexpertise spezifiziert, die im Rahmen einer Fall–Kontroll–Studie und damit auch für das WBS bzw. seinen Benutzer, relevant ist. Das Sachbereichs–, das Inferenz– und das Aufgabenwissen werden vorgestellt.

Das statische Sachbereichswissen (siehe Abschnitt 3.3.1) liefert zunächst eine ausführliche Beschreibung der Aufgabenbereiche, die im Organisationsmodell festgehalten worden sind, also eine Beschreibung der Datenerhebung, der Kontingenztafelanalyse sowie der Simulationsstudien und ihrer Auswertung. Durch diese Spezifikation des Sachbereichs sollen die zentralen Konzepte, ihre Eigenschaften und die Relationen dazwischen deutlich gemacht werden. Besondere Bedeutung bekommt diese Konzeptualisierung für die Auswertung von Simulationsstudien. Daher wird für diesen Bereich die Modellierung der Expertise systemgestützt weitergeführt und ausführlich in Kapitel 4 erläutert.

In 3.3.2 wird die sehr komplexe Inferenzstruktur für die Kontingenztafelanalyse anhand zweier beispielhaft vorgestellter Inferenzen erläutert.

In Abschnitt 3.3.3 wird das für das WBS relevante Aufgabenwissen spezifiziert. Dabei wird zunächst die Zerlegung der bereits erwähnten vier Hauptaufgabenbereiche in Unteraufgaben vorgestellt. Die Zerlegung für den Aufgabenbereich, der vom WBS unterstützt wird, also die Kontingenztafelanalyse, wird detailliert beschrieben und die Verteilung der Unteraufgaben an

die Agenten System und Benutzer vorgestellt. Es handelt sich in diesem Abschnitt also um eine Kombination aus dem fortgeführten Aufgabenmodell und dem Aufgabenwissen.

3.3.1 Sachbereichswissen (Domain Knowledge)

Im folgenden werden die vier für das WBS relevanten Aufgabenbereiche, die Datenerhebung, die Kontingenztafelanalyse, die Simulationsstudien und die Auswertung dieser Simulationsstudien beschrieben.

Design und Durchführung der Datenerhebung

Die Epidemiologie ist entstanden aus der Untersuchung von Epidemien, wie beispielsweise der Cholera. Heute beschäftigt man sich hier vor allem mit der Untersuchung der Verteilung von Krankheiten, physiologischen Variablen und sozialen Krankheiten in menschlichen Bevölkerungsgruppen sowie mit den Faktoren, die diese beeinflussen. Der Schwerpunkt der Forschung liegt damit in der Erfassung möglicher Ursache–Wirkungsbeziehungen, wobei sowohl positive als auch negative Assoziationen untersucht werden können.
Aus ethischen Gründen sind oft keine experimentellen Studien durchführbar. Es muß daher auf beobachtende Studien ausgewichen werden, die häufig ein aufwendigeres Design und eine komplexere Analyse erfordern. Zu den beobachtenden Studien zählen auch die Fall–Kontroll–Studien, mit deren Hilfe der Einfluß eines potentiellen Risikofaktors auf die Entstehung einer Krankheit analysiert werden kann.
Es gibt eine Reihe bekannter Untersuchungen, die in Form von Fall–Kontroll–Studien durchgeführt worden sind; beispielsweise wurden

- der Einfluß des Rauchens der Mutter auf angeborene Mißbildungen des Kindes (siehe [Kelsey et al, 1978]),
- der Einfluß von Allergien auf den plötzlichen Kindstod (siehe [Turner et al, 1975]),
- den Einfluß körperlicher Aktivität auf den Herzinfarkt (siehe [Hennekens et al, 1977]),
- der Einfluß von Strahlung auf Leukämie (siehe [Linos et al, 1980]),
- und der Einfluß von künstlichem Süßstoff auf Blasenkrebs (siehe [Hoover et al, 1980])

untersucht.
Fall–Kontroll–Studien sind im Gegensatz zu Kohortenstudien retrospektive Studien, d.h. die Probanden weisen die zu untersuchende Krankheit bereits auf, während in Kohortenstudien von gesunden Personen ausgegangen und die Entstehung einer Erkrankung beobachtet wird. Die Auswahl der Probanden hängt in diesen Studien im wesentlichen von der Exposition

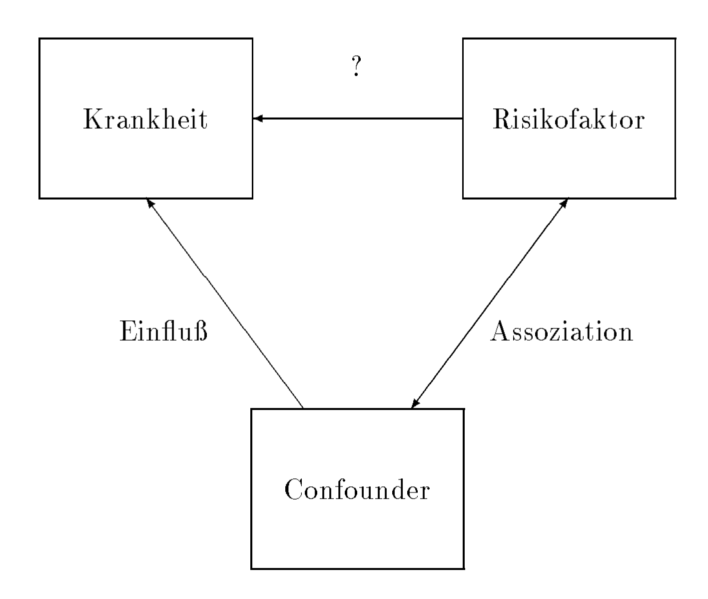


Abbildung 3.3: Assoziation von Krankheit, Risikofaktor und Confounder

bzgl. des Risikofaktors ab. Insbesondere bei seltenen Krankheiten mit hoher Latenzzeit ist eine Untersuchung mittels einer Kohortenstudie ungünstig.

Für die Datenerhebung in Fall–Kontroll–Studien benötigt man eine abgegrenzte Population erkrankter sowie nicht erkrankter Personen. Auch bei seltenen Krankheiten kann so eine ausreichend große Anzahl an Fällen sichergestellt werden. Aus der Population wird jeweils eine Stichprobe vom Umfang N_1 für die Fälle bzw. N_0 für die Kontrollen gezogen, $N_1, N_0 \in I\!N$. N_1 und N_0 sind Parameter, für die im Rahmen des Designs Werte festzulegen sind. Außerdem werden die Probanden hinsichtlich der Exposition untersucht. X_1 bzw. X_0 bezeichnet die Anzahl exponierter Personen in der Stichprobe der Fälle bzw. der Kontrollen, d.h. die Anzahl der Personen, die dem Risikofaktor ausgesetzt sind. Die Anzahl exponierter Personen wird als binomialverteilt angenommen, d.h. daß jede Person unabhängig von den anderen mit einer Wahrscheinlichkeit von p_1 exponiert ist, wenn sie zur Gruppe der Fälle gehört bzw. mit Wahrscheinlichkeit p_0, wenn sie zur Gruppe der Kontrollen gehört. Die Expositionswahrscheinlichkeiten sind also unbekannte Parameter und werden nicht durch das Design festgelegt.

Neben dem eigentlichen Risikofaktor gibt es dritte Größen, sog. Confounder, die einen Einfluß auf die Entstehung einer Krankheit haben können (siehe Abbildung 3.3). Ein Confounder ist eine diskrete oder stetige zufällige Einflußgröße, die für die interessierende Krankheit einen zusätzlichen Risikofaktor darstellt und mit dem betrachteten Risikofaktor assoziiert ist.

Der Einfluß des Confounders muß ausgeschaltet werden, da allein die Erhöhung der Erkran-

kungswahrscheinlichkeit bei Exposition durch den Risikofaktor bewertet werden soll. Berücksichtigt man vorhandene Confounder nicht, so kann das Ergebnis der Analyse verfälscht werden. Dies sei anhand des folgenden Beispiels illustriert:

> Man möchte den Einfluß des Alkoholkonsums auf die Entstehung eines Herzinfarkts untersuchen. Es ist bekannt, daß Rauchen einen Einfluß auf das Herzinfarktrisiko hat und auch mit dem Alkoholkonsum insofern in Verbindung steht, als daß starker Alkoholkonsum häufig einen starken Tabakkonsum nachsichzieht. Wird der Confounder Rauchen nicht berücksichtigt, so kann der Zusammenhang zwischen dem Alkoholkonsum und dem Herzinfarkt stark überschätzt werden.

Es gibt verschiedene Möglichkeiten, den Einfluß dieser Störgrößen zu eliminieren. Diese seien hier jedoch nur kurz aufgezählt. Durch 1:m Matching können Confounder bereits im Design der Studie berücksichtigt werden, indem man die Kontrollen so auswählt, daß zu jedem Fall m Kontrollen existieren, die bzgl. der Confounder die gleichen Merkmale aufweisen wie der entsprechende Fall. Andere Möglichkeiten bestehen darin, die Studie auf eine bzgl. der Confounder homogene Population einzuschränken, oder den Einfluß der Confounder erst auf der Ebene der Datenanlyse auszuschalten.
Confounder können aber auch erst später, nämlich auf der Ebene der Datenanalyse berücksichtigt werden (siehe unten).

Analyse von Fall–Kontroll–Studien

Um eine Aussage über den Grad der Assoziation zwischen dem potentiellen Risikofaktor und der Entstehung der Krankheit machen zu können, ist eine umfangreiche statistische Analyse der erhobenen Daten notwendig. Eine Möglichkeit bietet dazu die Durchführung einer Kontingenztafelanalyse.
Wurden Confounder im Design der Fall–Kontroll–Studie nicht berücksichtigt, so betrifft der erste Schritt der Analyse die Kontrolle der Confounder. Eine Methode hierzu ist die Schichtung der Daten bzgl. der Confoundervariablen. Eine Schichtung wird durchgeführt, indem man die Gruppe der Fälle bzw. der Kontrollen anhand der Ausprägungen der Confoundervariablen in disjunkte Teilmengen (Schichten) zerlegt. Handelt es sich bei den Confoundern nicht um diskrete Variablen oder sind deren Ausprägungen weiter zusammenzufassen, können zunächst Klassen gebildet werden.
Um einen gegebenen Datensatz zu schichten, besteht zum einen die Möglichkeit nicht relevante Kovariablen aus dem Datensatz auszusuchen. Dadurch können ggf. Schichten, die sich nur in den Ausprägungen dieser nicht mehr betrachteten Kovariablen unterscheiden, zusammengefaßt werden. Das Zusammenfassen dieser Schichten geschieht durch die Addition

		Altersgruppe									
		25–29		30–34		35–39		40–44		45–49	
Rauchen	OC	MI	Ctrl	MI	Ctrl	MI	Ctrl	MI	Ctrl	MI	Ctrl
Nein	Ja	0	25	0	13	0	8	1	4	3	2
	Nein	1	16	0	175	3	153	10	165	20	155
1–24	Ja	1	25	1	10	1	11	0	4	0	1
pro Tag	Nein	0	79	5	142	11	119	21	130	42	96
≥25	Ja	3	12	8	10	3	7	5	1	3	2
pro Tag	Nein	1	39	7	73	19	58	34	67	31	50

Tabelle 3.1: Beispiel für einen Ausgangsdatensatz

der Anzahlen für die exponierten und nicht exponierten Fälle und Kontrollen. Eine zweite Möglichkeit, einen Datensatz zu schichten, besteht darin, für bestimmte Kovariablen neue bzw. größere Intervalle für ihre Ausprägungen anzugegeben. Zur Verdeutlichung der Schichtung soll das folgende Beispiel dienen:

> Man betrachte die in der Tabelle 3.1 dargestellten Daten einer Fall–Kontroll–Studie (siehe [Shapiro et al, 1979]), die den Einfluß der Einnahme von oralen Kontrazeptiva auf den Herzinfarkt untersucht. Die erhobenen Kovariablen sind der Tabakkonsum und das Alter[1]. In diesem Datensatz liegen die Confounder „Rauchen", eingeteilt in drei Klassen, und das Alter, eingeteilt in fünf Klassen, vor.
>
> Soll der Tabakkonsum nun nicht mehr berücksichtigt werden, so erhält man den in Tabelle 3.2 gezeigten Datensatz. Hier wurde also die erste Methode der Schichtung angewendet. Dadurch, daß die Kovariable Rauchen nicht mehr berücksichtigt wird, muß zwischen der ersten, dritten und fünften bzw. zweiten, vierten und sechsten Zeile des Ausgangsdatensatzes nicht mehr unterschieden werden, da die restlichen Ausprägungen übereinstimmen. Diese Zeilen können damit jeweils zu einer Zeile

[1]In der Originalstudie wurden weitere Kovariablen betrachtet (Gewicht, Diabetes, Angina Pektoris u. a.), die hier zur Vereinfachung nicht einbezogen wurden.

	Altersgruppe									
	25–29		30–34		35–39		40–44		45–49	
OC	MI	Ctrl	MI	Ctrl	MI	Ctrl	MI	Ctrl	MI	Ctrl
Ja	4	62	9	33	4	26	6	9	6	5
Nein	2	134	12	390	33	330	65	362	93	301

Tabelle 3.2: Schichtung des Ausgangsdatensatzes durch Ignorieren des Risikofaktors Rauchen

		Altersgruppe					
		25–29		30–39		40–49	
Rauchen	OC	MI	Ctrl	MI	Ctrl	MI	Ctrl
Nein	Ja	0	25	0	21	4	6
	Nein	1	16	3	328	30	320
1–24	Ja	1	25	2	21	0	5
pro Tag	Nein	0	79	16	261	63	226
≥25	Ja	3	12	11	17	8	3
pro Tag	Nein	1	39	26	131	65	117

Tabelle 3.3: Intervallschichtung des Ausgangsdatensatzes bzgl. des Alters

zusammengefaßt werden, siehe Tabelle 3.2. Die Anzahlen ergeben sich durch das Aufsummieren der drei Werte in den ursprünglichen Schichten.

Eine Schichtung der Ursprungsdaten durch eine neue Intervallbildung für das Alter hat das in Tabelle 3.3 dargestellte Ergebnis, wenn folgende Einteilungen zugrunde gelegt werden: 25–29, 30–39 und 40–49.

Ebenfalls als eine neue Intervallbildung kann die Schichtung der Daten aufgefaßt werden, die den Tabakkonsum als eine dichotome Größe erfaßt, siehe Tabelle 3.4.

Wenn alle tatsächlichen Confounder bei der Schichtung berücksichtigt worden sind, beeinflussen sie die Ursache–Wirkungsbeziehung zwischen dem Risikofaktor und der Krankheit innerhalb einer Schicht nicht mehr.

		Altersgruppe									
		25–29		30–34		35–39		40–44		45–49	
Rauchen	OC	MI	Ctrl	MI	Ctrl	MI	Ctrl	MI	Ctrl	MI	Ctrl
Nein	Ja	0	25	0	13	0	8	1	4	3	2
	Nein	1	16	0	175	3	153	10	165	20	155
Ja	Ja	4	37	9	20	4	18	5	5	3	3
	Nein	1	118	12	215	30	177	55	197	73	146

Tabelle 3.4: Erfassung des Tabakkonsums als dichotome Größe

Gibt es eine große Anzahl von Confoundern, führt das häufig zu sehr kleinen Schichtumfängen. In solchen Fällen bieten sich als Alternativen zur Confounderkontrolle auf der Ebene der Datenanalyse logistische und loglineare Modelle an, die hier nicht nicht weiter betrachtet werden.
Der Einfluß von Confoundern, die im Rahmen der Datenerhebung nicht berücksichtigt wurden, d.h. über die keine Angaben erfaßt wurden, kann natürlich auch in der Datenanalyse nicht mehr korrigiert werden.

Die in der Fall–Kontroll–Studie beobachteten Daten werden in Tabellen, sog. 2x2–Kontingenztafeln (Vierfeldertafeln), dargestellt. Dabei gibt es vier Tabelleneinträge (Felder), jeweils ein Feld für exponierte Fälle, nicht exponierte Fälle, exponierte Kontrollen und nicht exponierte Kontrollen.
Werden zusätzlich Confounder berücksichtigt, so wird für jede Ausprägung dieser Confounder eine Kontingenztafel aufgestellt. Angenommen, der Confounder liegt in K Ausprägungen oder Kategorien vor, so ergeben sich K Schichten und somit K Kontigenztafeln, wobei die i–te Tafel, i=1,...,K, die in Tabelle 3.5 dargestellte Form hat. $a_i, \ldots, d_i$ ist die gebräuchliche Notation für die Zellen einer Kontingenztafel.
Es sei im folgenden zunächst der Fall K=1 diskutiert. Das Design von retrospektiven Studien macht es unmöglich, das in prospektiven Studien bekannte Maß für die Ursache–Wirkungsbeziehung, das relative Risiko, zu schätzen. Das relative Risiko P_1/P_0 vergleicht die bedingte Wahrscheinlichkeit, bei Exposition (E) an einer Krankheit (D) zu leiden, ($P_1 := P(D|E)$) mit der Erkrankungswahrscheinlichkeit, wenn keine Exposition (E^c) vorliegt ($P_0 := P(D|E^c)$). Es gibt damit den Faktor an, um den sich die Erkrankungswahrscheinlichkeit bei einer exponierten Person im Vergleich zu einer nicht exponierten erhöht.

Exposition

		ja	nein	Σ
Krank-	ja	a_i x_{1i}	b_i $N_{1i} - x_{1i}$	N_{1i}
heit	nein	c_i x_{0i}	d_i $N_{0i} - x_{0i}$	N_{0i}
	Σ	T_i	$N_i - T_i$	N_i

Tabelle 3.5: 2x2–Kontingenztafel

In Fall–Kontroll–Studien kann als Maß zur quantitativen Erfassung der vorliegenden Assoziation der Quotient (Ratio) aus der Chance (Odds) für das Auftreten der Krankheit, wenn der Risikofaktor vorliegt, also bei Exposition ($P_0/(1-P_0)$), und der entsprechenden Chance wenn er nicht vorliegt ($P_1/(1-P_1)$), herangezogen werden. Das Odds Ratio Ψ ergibt sich damit zu

$$\Psi = \frac{P_1(1-P_0)}{P_0(1-P_1)}.$$

Wenn die Erkrankungswahrscheinlichkeiten, also P_1 und P_0, sehr klein sind, approximiert Ψ das relative Risiko gut. Liegt eine Unabhängigkeit zwischen Risikofaktor und Krankheit vor, so nehmen sowohl Ψ als auch das relative Risiko den Wert eins an.
Da P_1 und P_0 unbekannt sind, muß das Odds Ratio Ψ geeignet geschätzt werden. Man betrachte dazu die Wahrscheinlichkeit im Falle einer Erkrankung exponiert zu sein ($p_1 := P(E|D)$), und die Wahrscheinlichkeit exponiert zu sein, wenn keine Erkrankung vorliegt ($p_0 := P(E|D^c)$).
Es kann nun das Bayessche Theorem angewendet werden, wonach gilt, daß

$$P(A_1|B) = \frac{P(B|A_1)P(A_1)}{P(B|A_1)P(A_1) + P(B|A_2)P(A_2)},$$

wobei A_1, A_2 und B stochastische Ereignisse und A_1 und A_2 komplementäre Ereignisse sind. Für die oben definierte bedingte Wahrscheinlichkeit P_1 gilt daher

$$P_1 = \frac{P(E|D)P(D)}{P(E|D)P(D) + P(E|D^c)P(D^c)} \text{ und somit } \frac{P_1}{1-P_1} = \frac{P(E|D)P(D)}{P(E|D^c)P(D^c)}.$$

Nach einer allgemeinen Formel für bedingte Wahrscheinlichkeiten ($P(A \cap B) = P(A|B)P(B)$) gilt

$$\frac{P_1}{1-P_1} = \frac{P(E \cap D)}{P(E \cap D^c)}.$$

Durch analoge Umformungen erhält man die Gleichungen

$$\frac{p_1}{1-p_1} = \frac{P(D \cap E)}{P(D \cap E^c)}, \; \frac{p_0}{1-p_0} = \frac{P(D^c \cap E)}{P(D^c \cap E^c)} \text{ und } \frac{P_0}{1-P_0} = \frac{P(E^c \cap D)}{P(E^c \cap D^c)}.$$

Daher gilt

$$\Psi = \frac{P_1(1-P_0)}{P_0(1-P_1)} = \frac{p_1(1-p_0)}{p_0(1-p_1)}.$$

Die Wahrscheinlichkeiten p_1 und p_0 können durch die beobachteten Häufigkeiten geschätzt werden, so daß sich das geschätzte Odds Ratio $\hat{\Psi}$ ergibt zu

$$\hat{\Psi} = \frac{X_1(N_0 - X_0)}{X_0(N_1 - X_1)}.$$

Wurden nun Confounder berücksichtigt und liegen daher mehrere Schichten vor, so kann für jede Schicht ein individuelles Odds Ratio berechnet werden. Für die i–te Schicht ergibt sich Ψ_i zu

$$\Psi_i = \frac{p_{1i}(1-p_{0i})}{p_{0i}(1-p_{1i})}, i = 1, \ldots, K.$$

Die individuellen Odds Ratios Ψ_i können durch die empirischen Odds Ratios $\hat{\Psi}_i$ geschätzt werden mit

$$\hat{\Psi}_i = \frac{X_{1i}(N_{0i} - X_{0i})}{X_{0i}(N_{1i} - X_{1i})}.$$

Der Effekt einer Schichtung wird deutlich, wenn man das auf den ungeschichteten Tafeln basierende $\hat{\Psi}$ mit den $\hat{\Psi}_i$'s vergleicht. Zur Illustration sei erneut auf das erste Beispiel zurückgegriffen:

> Ohne eine Schichtung hat die anfangs bereits nur inhaltlich erwähnte Untersuchung des Zusammenhangs von Alkoholkonsum und Herzinfarkt[2] die in Tabelle 3.6 dargestellten Ergebnisse erbracht, also ein mehr als zweimal höheres Herzinfarktrisiko bei Alkoholkonsum. Wird diese Kontingenztafel geschichtet bzgl. des Confounders Rauchen, so ergeben sich die Tafeln aus Tabelle 3.7. Die individuellen Odds Ratios weisen hier auf die Unabhängigkeit von betrachtetem Risikofaktor und Krankheit hin.

Geht man davon aus, daß der Confounder durch die Schichtung erfolgreich kontrolliert wurde, ist man nun daran interessiert, eine Aussage über den Zusammenhang zwischen dem Risikofaktor und der Krankheit zu treffen. Dazu kann u.a. ein sogenannter Unabhängigkeitstest

[2] Es handelt sich hierbei um einen hypothetischen Datensatz, der aus [Schlesselmann, 1982] übernommen worden ist.

		Alkoholkonsum		
		ja	nein	Σ
Herz-	ja	71	29	100
infarkt	nein	52	48	100
	Σ	123	77	200

$$\hat{\Psi} = 2.26$$

Tabelle 3.6: Gesamttafel

Nichtraucher

		Alkoholkonsum		
		ja	nein	Σ
Herz-	ja	8	22	30
infarkt	nein	16	44	60
	Σ	24	66	90

$$\hat{\Psi}_1 = 1.00$$

Raucher

		Alkoholkonsum		
		ja	nein	Σ
Herz-	ja	63	7	70
infarkt	nein	36	4	40
	Σ	99	11	110

$$\hat{\Psi}_2 = 1.00$$

Tabelle 3.7: Geschichtete Tafeln

durchgeführt werden. Das Ziel eines solchen Tests ist es, eine Aussage über die Unabhängigkeit von Risikofaktor und Krankheit zu machen. Als Ergebnis dieses Tests erhält man eine Teststatistik, die im Rahmen der Analyse von zentraler Bedeutung ist.

Neben der reinen Feststellung anhand des statistischen Tests, ob ein Zusammenhang vorliegt oder nicht, ist man natürlich auch an einer quantitativen Aussage über den Grad dieses Zusammenhangs interessiert. Dazu kann ein gemeinsames, von dem Einfluß des Confounders bereinigtes Odds Ratio geschätzt werden. Bei einer Reihe von Schätzern werden dazu die individuellen geschätzten Odds Ratios $\hat{\Psi}_i$ in geeigneter Weise zusammengefaßt. In dem Fall, daß der Confounder durch die Schichtung tatsächlich kontrolliert wurde, sind die theoretischen individuellen Odds Ratios alle gleich dem gemeinsamen Odds Ratio Ψ. Man spricht dann von der Homogenität der individuellen Odds Ratios. Die Homogenität der individuellen Odds Ratios kann nun durch eine Homogenitätsanalyse überprüft werden. Wird die Hypothese $H_0 := \Psi_1 = \ldots = \Psi_K = \Psi$ durch einen geeignet gewählten statistischen Test nicht verworfen, so ist dies ein Hinweis auf eine vorliegende Homogenität der individuellen Odds Ratios. In diesem Fall können die Informationen aus den K Kontingenztafeln zu einem Schätzer für das gemeinsame Odds Ratio kombiniert werden.

Zur Schätzung des gemeinsamen Odds Ratios sind eine Reihe von Verfahren bekannt. Es existieren iterative und nicht iterative Schätzer. Die aufwendiger zu berechnenden iterativen Schätzer beruhen auf dem Maximum–Likelihood–Prinzip. In der Praxis einfacher zu handhaben sind nicht–iterative Schätzer von Ψ. Bei diesen Schätzern handelt es sich um gewichtete Mittel der empirischen Odds Ratios $\hat{\Psi}_i$.

Hier seien die bekanntesten Punktschätzer von Ψ kurz vorgestellt. Der Woolf–Schätzer (siehe [Woolf, 1955]) summiert die logarithmierten geschätzten Odds Ratios und gewichtet sie mit den Inversen ihrer geschätzten Varianzen:

$$log\,\hat{\Psi}_W = \frac{\sum_{i=1}^{K}(1/X_{1i} + 1/(N_{1i} - X_{1i}) + 1/X_{0i} + 1/(N_{0i} - X_{0i}))^{-1} log\,\hat{\Psi}_i}{\sum_{i=1}^{K}(1/X_{1i} + 1/(N_{1i} - X_{1i}) + 1/X_{0i} + 1/(N_{0i} - X_{0i}))^{-1}}.$$

Der Mantel–Haenszel–Schätzer (siehe [Mantel, Haenszel, 1959]) ist ein gewichtetes artithmetisches Mittel der individuell geschätzten Odds Ratios $\hat{\Psi}_i$, wobei als Gewichte die Inversen ihrer geschätzten Varianzen unter der zusätzlichen Annahme $\Psi_1 = \ldots = \Psi_k = 1$ verwendet werden:

$$\hat{\Psi}_{MH} = \frac{\sum_{i=1}^{K} X_{1i}(N_{0i} - X_{0i})/N_i}{\sum_{i=1}^{K}(N_{1i} - X_{1i})X_{0i}/N_i}.$$

Da diese Schätzer verzerrt sind, ist es sinnvoll, ein Verfahren zur Verzerrungsreduktion anzuwenden, wie etwa das sogenannte Jackknife–Prinzip. Durch die Anwendung dieses Prinzips wird versucht, den Vorteil der einfacheren Berechenbarkeit der nicht–iterativen Schätzer mit

einer geeigneten Biaskorrektur zu kombinieren, um so zu verbesserten Eigenschaften zu gelangen. Zur Berechnung eines Jackknife–Schätzers wird jeweils eine Beobachtung aus dem Datensatz entfernt und mit diesem neuen Datensatz der Basisschätzer erneut berechnet. Man erhält durch eine bestimmte Linearkombination des Basisschätzers und der neu berechneten Schätzer die sogenannten Pseudowerte. Die Anzahl der Pseudowerte entspricht dabei gerade der Anzahl zu entfernender Beobachtungen. Das arithmetische Mittel dieser Pseudowerte liefert dann den Jackknife–Schätzer.

Im Rahmen einer geschichteten Kontingenztafelanalyse gibt es im wesentlichen zwei Ansätze zur Berechnung der Pseudowerte, die sich darin unterscheiden, was als eine Beobachtung bei der Berechnung der Pseudowerte angesehen wird. Der Ansatz von Breslow und Liang (siehe [Breslow, Day, 1980]), im weiteren Typ I genannt, betrachtet jede 2x2–Kontingenztafel als eine solche Beobachtung, die bei der Berechnung der Pseudowerte aus dem Datensatz entfernt wird.

Ein anderer Ansatz (siehe [Pigeot, 1989]), im weiteren mit Typ II bezeichnet, basiert darauf, daß jede binomialverteilte Zufallsvariable als Summe von unabhängig identisch Bernoulli–verteilten Zufallsvariablen aufgefaßt werden kann. Aus diesem Grund kann jede Beobachtung in jeder der vier Zellen einer Kontingenztafel als eine solche Beobachtung angesehen werden, die wiederum aus dem Datensatz entfernt wird.

Neben der beschriebenen Punktschätzung des gemeinsamen Odds Ratios ist auch eine Schätzung der Varianz und die Berechnung von Konfidenzintervallen durchzuführen. Die Varianz der Punktschätzung zeigt an, wie stark die wahren Odds Ratios um den Wert der Punktschätzung streuen. Ein Konfindenzintervall zum Niveau p ist ein Intervall, in dem das wahre Odds Ratio mit (1-p)–% Wahrscheinlichkeit liegt. Für die Schätzung der Varianz und der Konfidenzintervalle gibt es eine Reihe unterschiedlicher Methoden, auf die hier nicht näher eingegangen wird.

Die Liste der vorgeschlagenen Schätzer läßt sich beliebig erweitern. Ein Problem, das sich dem Anwender hier stellt, ist die Auswahl eines geeigneten Schätzers. Geeignete Kriterien für die Auswahl des Schätzers sind theoretisch natürlich zum einen die Abweichung der Schätzung vom wahren Odds Ratio und zum anderen die Variabilität des Schätzers. Da das wahre Odds Ratio aber unbekannt ist, kann diese Abweichung nicht berechnet werden.

Die Qualität eines Schätzers hängt von einer Reihe von Faktoren ab, unter anderem auch von der Datenlage der Fall–Kontroll–Studien. Die einzelnen Aspekte der Datenlage beeinflussen die verschiedenen Schätzer in unterschiedlicher Weise, d.h. es gibt durchaus Datensituationen, die günstig für einen bestimmten Schätzer sind, aber ungünstig für einen anderen. Betrachtet man die Konstruktionsprinzipien der einzelnen Schätzer genauer, so sind einige Gründe für solche unterschiedlichen Verhaltensweisen sofort ersichtlich. So können etwa bei Jackknife–

Schätzern vom Typ I Probleme auftreten, wenn nur wenige Tafeln (Schichten) vorhanden sind, da die Anzahl der Tafeln die Anzahl der Pseudowerte festlegt.
Eine genauere Untersuchung der finiten Eigenschaften, möglichst in Abhängigkeit der die Datenlage bestimmender Parameter, ist analytisch i.a. nicht möglich, so daß zur Beurteilung des finiten Verhaltens oft Simulationsstudien durchgeführt werden.

Design und Durchführung von Simulationsstudien

Simulationsstudien (Monte–Carlo–Studien) werden entworfen und durchgeführt, um die finiten Eigenschaften von Schätzern beurteilen zu können und um verschiedene, konkurrierende Verfahren zu vergleichen. Insbesondere neu entwickelte Schätzer können auf diese Weise detaillierter untersucht werden.
Im Rahmen des Simulationsdesigns zur Untersuchung von Schätzern des gemeinsamen Odds Ratios wird der wahre Wert des gemeinsamen Odds Ratios vorgegeben, so daß die Abweichung der Schätzer von diesem Wert beurteilt werden kann. Damit wird implizit die Homogenität der individuellen Odds Ratios vorausgesetzt. Weitere Parameter, die im Simulationsdesign festgelegt werden müssen, sind:

- die Anzahl der Tafeln K,
- die Anzahl der Fälle in der i–ten Schicht $N_{1i}, i = 1, \ldots, K$
- die Anzahl der Kontrollen in der i–ten Schicht $N_{0i}, i = 1, \ldots, K$
- die Expositionswahrscheinlichkeit der Kontrollen in der i–ten Schicht $p_{0i}, i = 1, \ldots, K$,

wobei z.B. bei der Festlegung der Anzahlen der Fälle und Kontrollen gleichzeitig auch das Verhältnis dieser Stichproben vorgegeben wird.
Das Design von Simulationsstudien orientiert sich an zwei asymptotischen Modellen, dem Large–Strata– und dem Sparse–Data–Modell, die bei der Diskussion der asymptotischen Eigenschaften der Odds Ratio Schätzer zugrunde gelegt werden. Dabei zeichnet sich das Large–Strata–Modell (Modell I) durch eine feste Anzahl von Tafeln aus, während die Stichprobenumfänge der Fälle und Kontrollen pro Tafel gegen unendlich streben. Im Sparse–Data–Modell (Modell II) sind dagegen die Stichprobenumfänge konstant, während die Anzahl der Tafeln gegen unendlich strebt. Bei der genauen Festlegung der Parameterkonstellationen einer Simulationsstudie sollten zwei Gesichtspunkte berücksichtigt werden. So sollten die Parameter zum einen so gewählt werden, daß damit möglichst interessante Aspekte aufgezeigt werden. Zum anderen sollten sie aber auch gerade typische, für die Praxis relevante Konstellationen abdecken.

Die Fragestellungen, die in einer solchen Simulationsstudie untersucht werden, sind beispielsweise:

- Gibt es eine Überlegenheit eines Schätzers in Abhängigkeit von der Anzahl der Schichten K oder der Größe des Odds Ratios Ψ?
- Gibt es eine Überlegenheit eines Schätzers in Abhängigkeit von N_{1i} und N_{0i}?
- Hängt die Eignung eines Schätzers mit den Expositionswahrscheinlichkeiten zusammen?
- Gibt es Kombinationen dieser Eigenschaften, die die Eignung des Schätzers beeinflussen?

Ein typisches Design für eine Monte–Carlo–Studie, die sich am Modell I, also dem Large–Strata–Modell orientiert, ist in der Tabelle 3.8 dargestellt. Aus der folgenden Beschreibung des Simulationsdesigns, die hier hauptsächlich aus [Pigeot, 1989] übernommen wurde, also vom Bereichsexperten stammt, können Hinweise auf die Kategorieneinteilungen (siehe 4.2.3) gewonnen werden.

Die vorliegende Simulationsstudie untersucht das Verhalten einiger nicht iterativer Schätzer für das gemeinsame Odds Ratio. Eine der zentralen Annahmen, die dem Simulationsaufbau zugrunde liegen, beinhaltet die Homogenität der Odds Ratios $\Psi_k, k = 1, \ldots, K$. Da sich die Studie am Modell I orientiert, sind die Anzahlen der 2x2–Tafeln relativ klein. Sie liegen in dem vorgestellten Design bei 2, 5 und 10. Für das gemeinsame Odds Ratio werden die Werte 1, 1.7, 3.5, 5 und 10 untersucht. Zur Begründung der Wahl dieser Werte wird in [Pigeot, 1989], Seite 101 folgendes angegeben:

> „... [Diese Festlegung des Parameters Ψ läßt Schlüsse darüber zu,] ob a. die Schätzer den Fall der Unabhängigkeit von Risikofaktor und Krankheit erkennen ($\Psi = 1$), b. ob sie auf einen leichten Zusammenhang reagieren ($\Psi = 1.7$) und c. ob ihre Verzerrung mit wachsendem Ψ zunimmt ($\Psi = 3.5$ und 5) bis hin zu dem extremen, eher unrealistischen Wert von $\Psi = 10$."

Für die Binomialwahrscheinlichkeiten $p_{0k}, k = 1, \ldots, K$, werden Werte gewählt, die nahe zusammenliegen und kleiner sind als 0.5 oder nahe zusammenliegen und um 0.5 zentriert sind oder weit auseinanderliegen und um 0.5 zentriert sind. Das Design dieser Simulationsstudie berücksichtigt, daß in vielen praktischen Situationen mehr Kontrollen als Fälle zur Verfügung stehen, indem die Anzahlen der Kontrollen stets größer sind als die der Fälle. In der Simulationsstudie sind Parameterkonstellationen vorhanden, in denen das Verhältnis der Anzahlen von Fällen und Kontrollen bei 1:3 liegt. Desweiteren gibt es bei gleichem Stichprobenumfang Situationen, in denen das Verhältnis für die verschiedenen Tafeln variiert. Das Design der Simulationsstudie umfaßt sowohl relativ kleine Stichprobenumfänge als auch auf das zehnfache

erhöhte Anzahlen von Fällen und Kontrollen, um den Einfluß der Stichprobengröße auf das Schätzverhalten zu unteruchen.

Die Simulationsstudie überprüft insgesamt 240 Parameterkonstellationen, die sich aus der Kombination der in der Tabelle 3.8 angegebenen Werte für die einzelnen Parameter ergeben. Für jede Konstellation werden 1000 Simulationsläufe durchgeführt. In jedem Lauf werden mittels eines Zufallszahlengenerators unter Zugrundelegung der jeweiligen Parameterkonstellation die Einträge in den Zellen der Kontingenztafeln als Realisationen entsprechend binomialverteilter Zufallszahlenvariabeln erzeugt. Für jeden Lauf wird dann anhand der einzelnen Verfahren das gemeinsame Odds Ratio geschätzt sowie die Standardabweichung und der Bias berechnet. Die 1000 Schätzergebnisse der Simulationsläufe werden im Anschluß gemittelt. Die Mittelwerte der geschätzten Odds Ratios können dann jeweils mit dem tatsächlichen Wert verglichen werden. Die Bewertungsmaße sind der Bias, also die Abweichung des geschätzten Werts vom tatsächlichen, der MSE, d.h. der mittlere quadratische Fehler, und die Standardabweichung. Dabei gilt, daß der MSE sich gerade ergibt als Summe aus der quadrierten Standardabweichung, also der Varianz und dem quadrierten Bias. Damit liefert die Standardabweichung keine zusätzliche Information.

Die Abbildung 3.4 zeigt den Aufbau der Dokumentation. Es wurden für jede Situation die Parameter beschrieben und die jeweiligen Ergebnisse der Schätzer bzgl. der Bewertungskriterien dokumentiert.

In der dargestellten Simulationsstudie wurden also sechs Schätzer verglichen, nämlich der Mantel–Haenszel und der Woolf–Schätzer, der Breslow–Liang–Schätzer, also der Typ I gejackknifte Mantel–Haenszel–Schätzer[3], der Jackknife–Schätzer Typ II (New Jackknife), basierend auf dem Mantel–Haenszel–Schätzer, ein weiterer Jackknife–Schätzer vom Typ II (New Jackknife 2), der aus dem logarithmierten Mantel–Haenszel–Schätzer berechnet wird sowie der Typ I gejackknifte Woolf–Schätzer. Die Schätzer werden im folgenden durch MH, W, BL, JK, JK_ii und W_JK abgekürzt.

Auswertung von Simulationsstudien

Die gewünschte Charakterisierung der Schätzer kann über die Dateneigenschaften erfolgen. Bei der Auswertung wird zunächst nach Abhängigkeiten gesucht, die von *einer* Dateneigenschaft auf eine Eignung eines Schätzers schließen. Solche Regeln sind zu bevorzugen, da sie sehr viele Datensituationen abdecken. Im weiteren können dann komplexere Abhängigkeiten untersucht werden.

Häufig werden die Simulationsergebnisse durch „Hinsehen“ ausgewertet. Es wird versucht, eine

[3] Die Berechnung dieses Jackknife–Schätzers basiert auf dem logarithmierten Mantel–Haenszel–Schätzer.

Parameter	**Werte**
K	2, 5, 10
Ψ	1.0, 1.7, 3.5, 5.0, 10.0
$N_{1i} : N_{0i}$	20:60, 30:90
$i = 1, 2$	20:70, 30:80
$N_{1i} : N_{0i}$	20:60, 25:75, 30:90, 35:105, 40:120
$i = 1, \ldots, 5$	20:55, 25:85, 30:80, 35:95, 40:135
$N_{1i} : N_{0i}$	20:60, 20:60, 25:75, 25:75, 30:90, 30:90, 35:105, 35:105, 40:120, 40:120
$i = 1, \ldots, 10$	20:55, 20:60, 25:85, 25:80, 30:80, 30:85, 35:95, 35:90, 40:135, 40:135
p_{0i}	0.20, 0.30
$i = 1, 2$	0.20, 0.80
	0.45, 0.55
p_{0i}	0.20, 0.22, 0.25, 0.28, 0.30
$i = 1, \ldots, 5$	0.20, 0.40, 0.50, 0.60, 0.80
	0.45, 0.48, 0.50, 0.52, 0.55
p_{0i}	0.20, 0.22, 0.24, 0.26, 0.28, 0.30, 0.32, 0.34, 0.36, 0.38
$i = 1, \ldots, 10$	0.20, 0.25, 0.30, 0.40, 0.45, 0.55, 0.60, 0.70, 0.75, 0.80
	0.40, 0.42, 0.44, 0.46, 0.48, 0.52, 0.54, 0.56, 0.58, 0.60

Zusätzliche Werte für $p_{0i}, i = 1, \ldots, K$

K=2	0.05, 0.15, 0.05, 0.95
K=5	0.05, 0.07, 0.09, 0.11, 0.13, 0.05, 0.25, 0.50, 0.75, 0.95
K=10	0.05, 0.07, 0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.21, 0.23
	0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95

Tabelle 3.8: Simulationsdesign

```
1-te Situation
==============

Dokumentation
==============

Anzahl Situationen:    1000
Anzahl Tafeln:            2
Psi:                      1.000
Tafel      N1       N2       P1         P2
-------------------------------------------------
   1       20       60       0.20000    0.20000
   2       30       90       0.30000    0.30000

Ergebnisse
==========
Anzahl der verworfenen Situationen:     6
Anteil der verworfenen Situationen: 7.494E-04
           Schaetzer    Mean       Std        MSE        Bias      Neg
------------------------------------------------------------------------
Mantel-Haenszel         1.05524    0.01385    0.19475    0.05524   0
Woolf                   1.07759    0.01393    0.19994    0.07759   0
Breslow-Liang           1.09289    0.01501    0.23391    0.09289   0
New Jackknife           0.99379    0.01281    0.16422   -0.00621   0
New Jackknife 2         1.07214    0.01362    0.19074    0.07214   0
Woolf-Jackknife         1.15193    0.01733    0.32350    0.15193   0
```

Abbildung 3.4: Dokumentation der Simulationsstudien

Beschreibung von Situationen mit den gleichen Eigenschaften für alle oder bestimmte Schätzer zu finden, die diese vom Rest der Situationen abgrenzt. Durch die Vielzahl der variierten Parameter ist dies recht problematisch. Hier hat sich gezeigt, daß komplexere Abhängigkeiten auf diese Weise schwer zu entdecken sind. So ist es schon sehr aufwendig, für alle Kombinationen den Einfluß von zwei Dateneigenschaften auf die Schätzergebnisse zu kontrollieren. In der Praxis werden daher häufig Vermutungen, die aufgrund von Erfahrungen oder aus theoretischen Überlegungen entstanden sind, anhand der Simulationsergebnisse überprüft oder besonders aufällige Ergebnisse untersucht, z.B. besonders schlechtes oder gutes Abschneiden eines Schätzers.
Eine solche Auswertung eignet sich deshalb dazu, einen neuen Schätzer grob zu bewerten, hier beispielsweise nachzuweisen, daß die Biaskorrektur durch das Jackknife–II–Prinzip erfolgreich war, aber um eine „vollständige“ Charakterisierung der Schätzer zu erhalten, sind weitergehende Auswertungsmethoden anzuwenden. Dabei sind in der Statistik eine Reihe von Methoden bekannt, die hier zum Einsatz kommen könnten, wie z.B. :

- Clusteranalysen,
- Korrelationskoeffizienten und
- Korrespondenzanalysen.

Im Rahmen dieser Diplomarbeit wird ein anderer Ansatz verfolgt, der auf dem Einsatz von Methoden aus der Künstlichen Intelligenz basiert. Die zentrale Idee dieses Ansatzes besteht darin, die gewünschte Charakterisierung der Schätzer maschinell lernen zu lassen. Das verwendete Lernverfahren ist in das Modellierungssystem MOBAL (siehe [Morik et al, 1993]) integriert. Mit diesem System wird aus den Simulationsstudien ein Beispielset konstruiert, aus dem Regeln über das Verhalten der Schätzer gelernt werden. Eine ausführliche Beschreibung dieses Ansatzes ist in Kapitel 4 zu finden.
Dieser Ansatz ermöglicht es, eine ähnliche Vorgehensweise wie bei der gerade beschriebenen Auswertung zu benutzen. So können die Ergebnisse, die zuvor schon durch diese Auswertung erzielt worden sind, reproduziert bzw. widerlegt werden. Der Vorteil besteht darin, eine umfangreichere Regelmenge erhalten zu können, indem eine vollständige Überprüfung der Kombinationen von bis zu fünf Dateneigenschaften durchgeführt wird.

3.3.2 Inferenzwissen (Inference Knowledge)

In diesem Abschnitt wird das für das WBS relevante Inferenzwissen vorgestellt. Es müssen auf der Inferenzebene eine Reihe von primitiven Inferenzen spezifiziert werden, die auf der

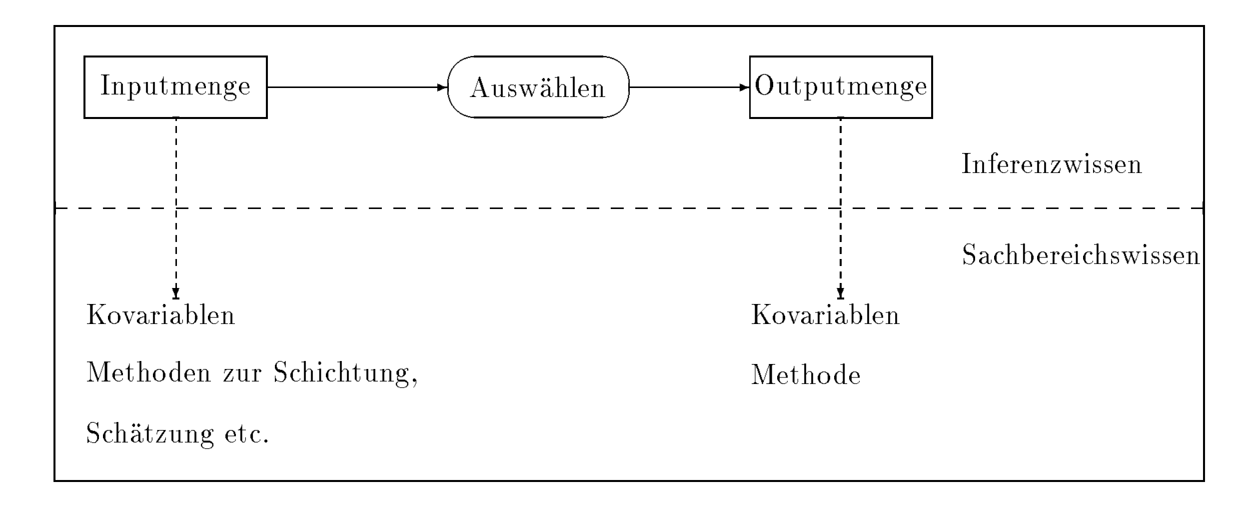


Abbildung 3.5: Die primitive Inferenz „Auswählen“

darüberliegenden Aufgabenebene zur Ausführung der Unteraufgaben der Kontingenztafelanalyse verwendet werden. Die Inferenzebene ist so komplex, daß sie hier nur beispielhaft beschrieben werden soll. Eine vollständige Liste aller Inferenzen ergibt sich aus der Aufgabenstruktur (Task Structure) für die einzelnen Unteraufgaben aus dem Abschnitt 3.3.3.

Im folgenden werden nun die Inferenzschritte „Auswählen“ und „Ausführen“ spezifiziert, die im Aufgabenwissen jeweils mehrfach verwendet werden.

Die Knowledge Source „Auswählen“ (siehe Abb. 3.5) greift auf die zwei Metaklassen „Inputmenge“ und „Outputmenge“ zu. Die Objekte des Sachbereichs, die diesen Metaklassen angehören, unterscheiden sich nicht: es sind jeweils die Kovariablen und die verschiedenen Methoden für die Analyseschritte. Durch diese Inferenz wird beispielsweise vom Benutzer eine Methode für einen Homogenitätstest ausgewählt, oder es werden die relevanten Kovariablen für die weitere Analyse bestimmt.

Die Wissensquelle „Ausführen“ (siehe Abb. 3.6) besitzt zwei Input–Metaklassen, die Methoden und die Daten. Die Metaklasse Methoden umfaßt die Methoden zur Schichtung der Daten, die Homogenitätstests, die diversen Schäzer, die Unabhängigkeitstests sowie die Methoden zur Ziehung einer Zufallsauswahl aus den Daten. Die Output–Metaklasse wird als Ergebnis bezeichnet. Sie enthält die folgenden Objekte des Sachbereichs: die geschichteten Daten der Fall–Kontroll–Studie, die p–Werte bzw. Teststatistiken, die geschätzten Werte wie z.B. das geschätzte Odds Ratio, dessen Varianz etc. sowie die ausgewählte Stichprobe (Pilotstudie) aus den Fall–Kontroll–Studiendaten.

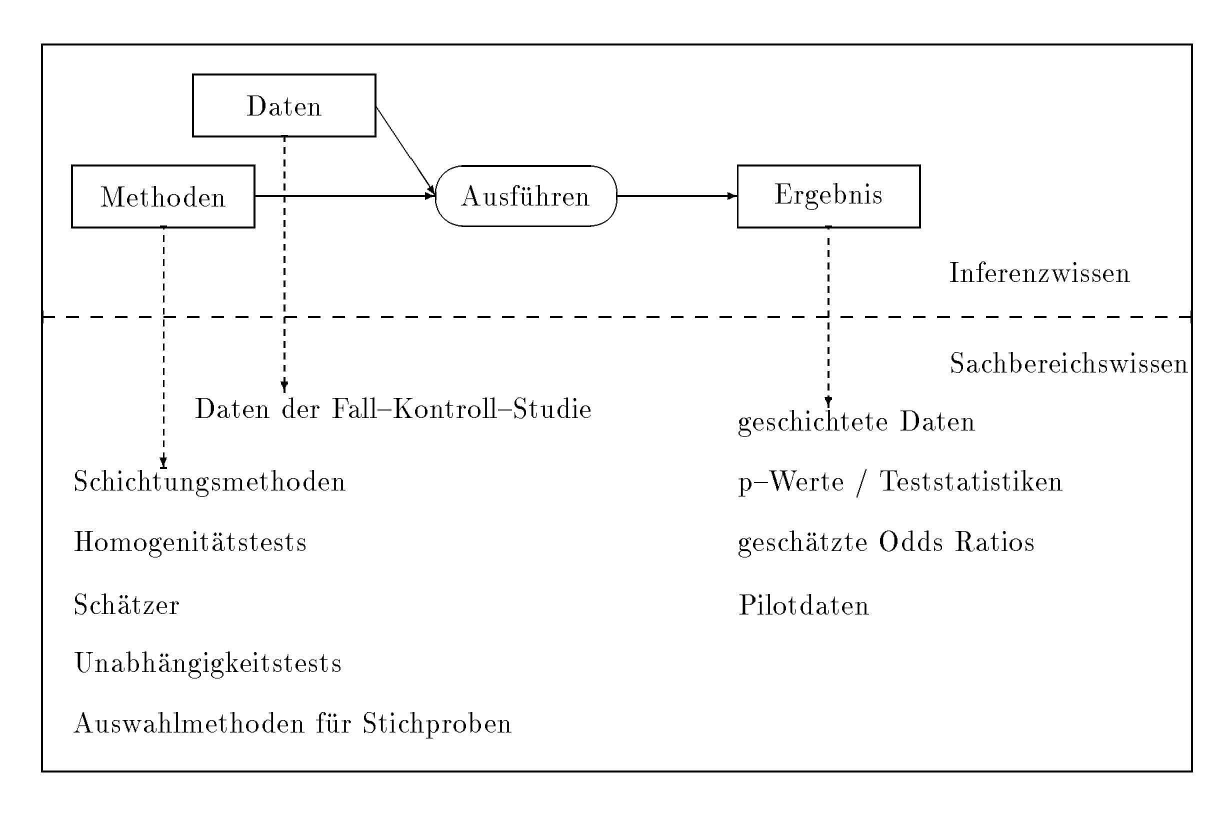


Abbildung 3.6: Die primitive Inferenz „Ausführen“

3.3.3 Aufgabenwissen (Task Knowledge)

Dieser Abschnitt führt, wie bereits erwähnt, eine Zerlegung der Hauptaufgabenbereiche in ihre Unteraufgaben im Sinne eines Aufgabenmodells durch. Da für ein besseres Verständnis dieser Zerlegung die zentralen Konzepte des Sachbereichs bekannt sein sollten, erfolgt eine detaillierte Darstellung erst an dieser Stelle zusammen mit der Spezifikation des Aufgabenwissens. Damit wird also jeder Aufgabenbereich durch die Spezifikation des Ziels, des In– und Outputs und der Aufgabenstruktur dargestellt. Der Input entspricht der Information, die zum Erreichen des Ziels benötigt wird. Output und Ziel sind in den meisten Fällen identisch. Die Aufgabenstruktur teilt die Aufgaben in eine hier zumeist sequentielle Abfolge von Unteraufgaben ein.

Datenerhebung

Die Datenerhebung kann aufgeteilt werden in das Design der Fall–Kontroll–Studie und die eigentliche Datensammlung. Diese Unteraufgaben sind klar getrennt und werden sequentiell abgearbeitet. Eine weitergehende Aufgabenunterteilung ist hier nicht interessant, da diese Aufgaben vom wissensbasierten System nicht unterstützt werden sollen.

Task *Datenerhebung*

Ziel: Sammlung der Daten

Input: Design–Parameter

Output: Fall–Kontroll–Studiendaten

Control-Terms:

Werte: *Werte für Design–Parameter wie Fallanzahlen etc.*

Confounder: *Größen, die die Ursache–Wirkungsbeziehung beeinflussen und daher erhoben werden müssen*

Klasseneinteilungen: *Klassen, in denen die Confounder erhoben werden sollen*

Aufgabenstruktur

Datenerhebung(Design–Parameter→Fall–Kontroll–Studiendaten)=

Design(Design–Parameter→Werte für Design–Parameter+Confounder+ Klasseneinteilungen für Confounder)

Sammlung(Werte für Design–Parameter+Confounder→Fall–Kontroll–Studiendaten)

Kontingenztafelanalyse

Bei der Kontingenztafelanalyse sind die Schichtung, die Homogenitätsanalyse, die Schätzung des gemeinsamen Odds Ratios und die Unabhängigkeitstests zu unterscheiden.

Dabei ist zu beachten, daß die Durchführung der Unteraufgaben relativ unabhängig voneinander erfolgt in der Weise, daß die Durchführung einer Aufgabe die Durchführung einer anderen nicht bedingt. So kann beispielsweise, auch wenn dieses Vorgehen wenig sinnvoll zu sein scheint, auf eine Schichtung oder auf eine Homogenitätsanalyse verzichtet werden. Desweitern kann z.B. die Durchführung eines Unabhängigkeitstests auch vor der Schätzung des Odds Ratios erfolgen.

Task *Kontingenztafelanalyse*

Ziel: Aufschluß über den Zusammenhang von Risikofaktor und Krankheit

Input: Methoden, Daten: Daten einer Fall–Kontroll–Studie

Output: Odds Ratio, Testentscheidungen

Control-Terms:

Methoden 1: *Methoden zur Schichtung*

Methoden 2: *Methoden zur Homogenitätsanalyse*

Methoden 3: *Methoden zur Schätzung des Odds Ratios*

Methoden 4: *Methoden für Unabhängigkeitstests*

p–Wert 1: *Wert zwischen 0 und 1, Maß für die Homogenität der individuellen Odds Ratios*

p–Wert 2: *Wert zwischen 0 und 1, Maß für die Unabhängigkeit von Risikofaktor und Krankheit*

Aufgabenstruktur

Kontingenztafelanalyse(Daten→Odds Ratio)=

Schichtung(Daten+Methoden 1→Daten)

Homogenitätsanalyse(Daten+Methoden 2→p–Wert 1)

Odds Ratio Schätzung(Daten+Methoden 3+p–Wert 1→Odds Ratio)

Unabhängigkeitstest(Daten+Methoden 4→p–Wert 2)

Da alle Aufgaben mit Unterstützung des wissensbasierten Systems durchgeführt werden, folgt nun eine genauere Vorstellung der einzelnen Subtasks.

Schichtung

Die Durchführung der Unteraufgabe Schichtung ist, wie bereits erwähnt, optional, kann aber auch wiederholt durchgeführt werden. Input der Schichtung sind die Fall–Kontroll–Studiendaten sowie Methoden zur Schichtung. Der Output sind die geschichteten Daten.

Ein erster Auswahl–Task, der also auf die oben beschriebene primitive Inferenz zurückgreift, betrifft die Kovariablen. Es muß vom Benutzer angegeben werden, welche Kovariablen bei der Schichtung berücksichtigt werden sollen. Die nicht berücksichtigten Kovariablen werden für die weitere Analyse gestrichen.

Der zweite Subtask ist die Auswahl einer der im Sachbereichswissen beschriebenen Methoden zur Schichtung. Nur falls die Intervallschichtung ausgewählt wurde, müssen im nächsten Schritt Intervalle für die angegebenen Kovariablen bestimmt werden.

Das Ausführen der Schichtung erzeugt einen neuen Datensatz, der im letzten Schritt anhand eines geeigneten Homogenitätstests überprüft wird. Das Ziel der Überprüfung ist es, festzustellen, ob der betrachtete Confounder durch die Schichtung tatsächlich kontrolliert werden konnte. Ist das nicht der Fall, so kann der ursprüngliche Datensatz erneut geschichtet werden, indem ggf. Schichten zusammengefaßt oder neu gebildet werden bzw. mehr Confounder berückücksichtigt werden.

Task *Schichtung*

Ziel: Einfluß von Confoundern ausschalten

Input: Methoden, input–Daten

Output: Output–Daten

Control-Terms:

Input–Daten: *geschichtete oder ungeschichtete Daten*

Output–Daten: *geschichtete Daten*

Kovariablen: *Confoundervariablen, gehören zum Datensatz*

Ausprägungen: *Werte für Kovariablen*

Intervalle: *Klassen für Werte der Kovariablen*

Entscheidung: *Entscheidung, ob die Kontrolle der Confounder als erfolgreich angesehen wird*

Task structure

Schichtung(Daten+Methoden→Daten)=

Auswählen(Kovariablen→Kovariablen)

Auswählen(Methoden→Methode)

Klassifizieren(Ausprägungen→Intervalle)

Ausführen(Daten+Intervalle+Methode→Daten)

Überprüfung(Daten→Entscheidung)

Homogenitätsanalyse

Das Ziel der Homogenitätsanalyse ist es, eine Aussage über die Homogenität der Odds Ratios zu erhalten. Als Maß hierfür wird der p–Wert eines geeignet ausgewählten statistischen Tests berechnet. Dieser Wert wird auf Wunsch erklärt, d.h. es wird ein Vorschlag über die Annahme oder Ablehnung der Homogenität gemacht.[4] Die Homogenitätsanalyse wird im Rahmen des Systems nur explorativ genutzt, d.h., daß der berechnete p–Wert als Indikator für oder gegen die Annahme der Homogenität angesehen wird. Dabei sprechen große p–Werte für die Homogenitätsannahme. Die Testentscheidung ist damit auch nicht zwingend für den Benutzer, d.h. selbst bei abgelehnter Homogenitätsannahme kann ein gemeinsames Odds Ratio geschätzt werden. Der dritte Task, die Entscheidung des Benutzers, muß also nicht dem Vorschlag des Systems entsprechen.

[4] Falls der p–Wert kleiner als 0.1 ist, wird vorgeschlagen, die Homogenitätshypothese abzulehnen.

Task *Homogenitätsanalyse*

Ziel: Aussage über die Homogenität der individuellen Odds Ratios

Input: Methoden, Daten

Output: Teststatistik, p–Wert

Control-Terms:

Vorschlag: *Vorschlag über die Annahme oder Ablehnung der Homogenität*

Entscheidung: *Annahme oder Ablehnung der Homogenität*

Task structure

Homogenitätsanalyse(Daten+Methoden→Teststatistik+p–Wert)=

Auswählen(Methoden→Methode)

Ausführen(Daten+Methode→Teststatistik)

Standardisierung(Teststatistik→p–Wert)

Erklären(p–Wert→Vorschlag)

Entscheiden(p–Wert+Vorschlag→Entscheidung)

Odds Ratio Schätzung

Die Odds Ratio Schätzung kann als zentrale Aufgabe der Kontingenztafelanalyse angesehen werden. Ihr Output ist das geschätzte Odds Ratio. Als Input benötigt dieser Task die geschichtenten Daten der Fall–Kontroll–Studie und Methoden, also die Schätzer (siehe Abschnitt 3.3.1). Die Entscheidung des Homogenitätstests geht indirekt als Input in diese Aufgabe ein.

Die drei Tasks, die hier zusammen die Odds Ratio Schätzung ausmachen, sind die Pilotstudie, eine Methodenauswahl und die Ausführung der Schätzung. Dabei bedarf die Pilotstudie weiterer Erläuterung durch eine Zerlegung in ihre Unteraufgaben.

Task *Odds Ratio Schätzung*

Ziel: Aussage über den Zusammenhang von Risikofaktor und Krankheit

Input: Daten, Methoden, ggf. p–Wert

Output: Odds Ratio

Control-Terms:

Pilotstudiendaten: *Teilmenge der Fall–Kontroll–Studiendaten*

Methodenempfehlung: *geeigneter Schätzer für die vorliegende Datensituation*

Task structure

Odds Ratio Schätzung(Daten+Methoden+p–Wert→Odds Ratio)=

Pilotstudie(Daten→Pilotstudiendaten+Methodenempfehlung)

Auswählen(Methoden→Methode)

Ausführen(Daten+Methode→Odds Ratio)

Das Ziel, das mit der Durchführung einer Pilotstudie verfolgt wird, ist die Entscheidungsunterstützung für die Methodenauswahl in der Fall–Kontroll–Studie.

Wichtige Einflußfaktoren auf die Schätzergebnisse sind die Größe des Odds Ratios und die Größe der Expositionswahrscheinlichkeiten. Da beide Größen aber unbekannt sind, kann nur auf eine Schätzung zurückgegriffen werden. Problematisch ist hier, daß so die Auswahl des Schätzers von der Schätzung selbst abhängt. Eine Lösung dieses Problems besteht in der Durchführung einer Pilotstudie. Eine Pilotstudie ist, wie bereits kurz erwähnt, eine Fall–Kontroll–Studie kleineren Umfangs unter ansonsten gleichen Bedingungen, d.h. die Design–Parameter Tafelanzahl und das Verhältnis von Kontrollen und Fällen bleiben unverändert gegenüber der Fall–Kontroll–Studie. Daten für Pilotstudien lassen sich am einfachsten dadurch erhalten, daß man einen kleinen Teil der Fall–Kontroll–Studiendaten für diese Studie verwendet. Sie können dann nicht mehr für die eigentliche Auswertung der Fall–Kontroll–Studie verwendet werden. Damit besteht der offensichtliche Nachteil dieses Vorgehens in der Reduktion der Datenbasis.

Es gibt mehrere Methoden zur Erzeugung eines Pilotdatensatzes: z.B. die einfache und die geschichtete Zufallsauswahl. Die einfache Auswahl zieht eine Stichprobe mit einem vorgegebenen Umfang aus den Originaldaten. Die geschichtete Auswahl zieht einen bestimmten Anteil aus jeder Schicht der Originaldaten. Als Output entstehen jeweils zwei neue Datensätze: die Pilotdaten und die neuen Fall–Kontroll–Studiendaten, aus denen die Pilotdaten entfernt worden sind.

Die Pilotdaten werden mit einer geeigneten Methode analysiert. Die Analyse umfaßt hier die Schätzung des gemeinsamen Odds Ratios und die Berechnung der Expositionswahrscheinlich-

keiten für diese Daten.

Sowohl diese Ergebnisse als auch die Werte für die Design–Parameter der Fall–Kontroll–Studie werden im Anschluß klassifiziert. Diese Klassifizierung wird vom System vorgenommen und teilt die Dateneigenschaften in feste Kategorien ein. Die Kategorien entsprechen denen, die im Rahmen der Simulationsstudienauswertung festgelegt wurden (vgl. dazu Abschnitt 3.3.1 bzw. Kapitel 4).

Der Subtask Assistieren erzeugt die den klassifizierten Dateneigenschaften entsprechende Methodenempfehlung. Dazu wird auf die in das System integrierte Regelmenge zurückgegriffen. Auf Wunsch wird die Methodenempfehlung erläutert, indem die Eignung der Methode, die abgeleitet wurde, mitgeteilt wird und die zugrundeliegenden klassifizierten Dateneigenschaften abgerufen werden können.

Task *Pilotstudie*

Ziel: Entscheidungsunterstützung bei der Methodenauswahl

Input: Daten

Output: Pilotstudiendaten, Methodenempfehlung

Control-Terms:

Pilot Odds Ratio: *Odds Ratio geschätzt durch den Pilotdatensatz*

Pilot Expositionswahrscheinlichkeiten: *Expos. berechnet aus den Pilotdaten*

Design Parameter: *Parameter, die im Design der FKS festgelegt wurden*

klassifizierte Daten: *Bewertung für Größe des Odds Ratios,*
der Expos. und der Werte für die Design Parameter

Regelmenge: *Regeln, die aus der Auswertung der SimSt hervorgegangen sind,*
schließen von Dateneigenschaften auf die Eignung der Schätzer

Erklärung: *Aussagen über Eignung des empfohlenen Schätzers*
und zugrundeliegende Dateneigenschaften

Task structure

Pilotstudie(Daten→Pilotstudiendaten+Methodenempfehlung)=

Auswählen(Daten→Pilotstudiendaten)

Analysieren(Pilotstudiendaten+Methode→Pilot–Odds Ratio+
Expositionswahrscheinlichkeiten)

Klassifizieren(Pilot–Odds Ratio+Expos.+Design–Parameter→klassifizierte Daten)

Assistieren(Regelmenge+klassifizierte Daten→Empfehlung)

Erklären(Empfehlung→Erklärung)

Unabhängigkeitstest

Das Ziel eines Unabhängigkeitstests ist es, eine Aussage darüber zu treffen, ob das gemeinsame wahre Odds Ratio von eins verschieden ist[5] ist, zu treffen. Als Input dienen unterschiedliche statistische Tests und die Daten der Fall–Kontroll–Studie. Der Output ist ähnlich wie für die Homogenitätsanalyse, ein Maß, das die Unabhängigkeit zwischen Risikofaktor und Krankheit beschreibt. Diese Teststatistik wird dann noch geeignet standardisiert, so daß man den sog. p–Wert erhält. Anders als bei der Homogenitätsanalyse wird dem Anwender eine Testentscheidung hier vom System mitgeteilt.

Task *Unabhängigkeitstest*

Ziel: Aussage über Unabhängigkeit zwischen Risikofaktor und Krankheit

Input: Methoden, Daten: ungeschichtet bzw. bereits geschichtet

Output: Testentscheidung: $\Psi = 1$ oder $\Psi \neq 1$

Task structure

Unabhängigkeitstest(Daten+Methoden→Testentscheidung)=

Auswählen(Methoden→Methode)

Ausführen(Daten+Methode→Teststatistik)

Standardisierung(Teststatistik→p–Wert)

Entscheiden(p–Wert→Testentscheidung)

Die Abhängigkeiten zwischen den gerade beschriebenen Unteraufgaben der Kontingenztafelnanalyse werden in Abb. 3.7 dargestellt.

Simulationsstudien

Mithilfe der Simulationsstudien werden die finiten Eigenschaften der Methoden (Schätzer) untersucht, die in der Kontingenztafelanalyse verwendet werden. Als Input werden die zu untersuchenden Methoden benötigt. Außerdem muß eine Entscheidung über das zugrundeliegende Modell (siehe 3.3.1) getroffen werden. Der Output besteht aus den Parameterkonstellationen, d.h. den Werten für die Designparameter der Simulationsstudie und den Schätzergebnissen in den jeweiligen Situationen. Zu beachten ist hier, daß sich die Parameter für das Design der Simulationsstudien zum Teil von den Designparametern der Fall–Kontroll–Studie unterscheiden. So gehört bei den Simulationsstudien auch das Odds Ratio zu den im Design festgelegten Größen.

[5] Ein Odds Ratio von eins bedeutet Unabhängigkeit von Risikofaktor und Krankheit.

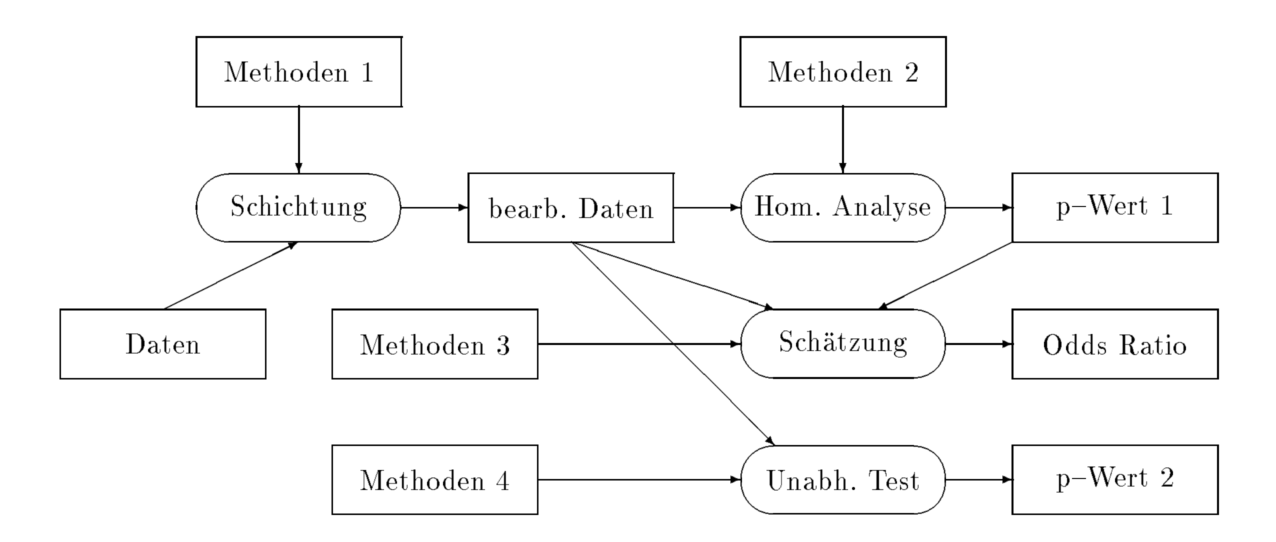


Abbildung 3.7: Aufgabenstruktur der Kontingenztafelanalyse

Die Subtasks für die Durchführung von Simulationsstudien sind das Design der Studien, die Ausführung der Simulation und ihre Dokumentation. Diese Subtasks bauen aufeinander auf und erfolgen daher streng sequentiell.

Task *Simulationsstudien*

Ziel: Untersuchung der Methoden (Schätzer)

Input: Methoden, Modell

Output: Parameterkonstellationen, Schätzergebnisse, Dokumentation

Aufgabenstruktur

Simulationsstudien(Methoden+Modell→Parameterkonstellationen)=
Schätzergebnisse+Dokumentation

Design (Modell→Parameterkonstellationen)
Ausführen(Parameterkonstellationen+Methoden→Schätzergebnisse)
Dokumentation(Parameterkonstellationen+Schätzergebnisse→Dokumentation)

Auswertung von Simulationsstudien

Die Auswertung der Simulationsstudien wird im Kapitel 4 ausführlich beschrieben. In diesem Abschnitt folgt nur eine knappe Zusammenfassung in der KADS–Methodologie.
Das Ziel der Auswertung von Simulationsstudien ist es, eine Aussage über das datenabhängige Verhalten der einzelnen Schätzer machen zu können. Durchgeführt wird diese Aufgabe

hier mithilfe des maschinellen Lernens. Die Simulationsstudien, genauer ihre Dokumentation, dienen als Input. Der Output ist hier eine Regelmenge, die die Schätzer charakterisiert.

Die Subtasks der Auswertung von Simulationsstudien bei Verwendung des in Kapitel 4 beschriebenen wissensbasierten Ansatzes sind die Modellierung, also die Generierung von Beispielen, der Entwurf von Lernläufen und ihre Durchführung sowie die Bewertung der Ergebnisse.

Task *Auswertung von Simulationsstudien*

Ziel: Aussage über das datenabhängige Verhalten der einzelnen Schätzer

Input: Simulationsdaten, Dokumentation

Output: Regelmenge

Aufgabenstruktur

Auswertung von Simulationsstudien(Simulationsdaten+Dokumentation→Regelmenge)=

Modellierung (Simulationsdaten→Faktenmenge)

Entwurf(Faktenmenge+Metaprädikate+Zielprädikate+Akzeptanzkriterien→ Lernläufe)

Durchführung(Lernläufe→Regeln)

Auswahl/Bewertung(Regeln+Bewertungskriterien+Bewertung durch XP→ Regelmenge)

Die Modellierung kann in weitere Unteraufgaben zerlegt werden, nämlich in die Bestimmung der relevanten Dateneigenschaften, die Berechnung von abgeleiteten Dateneigenschaften, das Bilden von Kategorien und die Repräsentation. Das Ziel der Modellierung ist der Aufbau einer Faktenmenge, aus der mithilfe des Systems MOBAL/RDT die oben beschriebenen Regeln gelernt werden können.

Task *Modellierung*

Ziel: Aufbau einer Wissensbasis

Input: Simulationsdaten

Output: Wissensbasis

Control-Terms:

Kategorien: *Kategorien für die Dateneigenschaften und Bewertungskriterien*

Aufgabenstruktur

Modellierung(Simulationsdaten→Faktenmenge)=

Auswahl(Simulationsdaten→Dateneigenschaften+Bewertungskriterien)

Klassifikation(Dateneigenschaften+Bewertungskriterien→Kategorien)

Repräsentation(Kategorien+Simulationsdaten→Wissensbasis)

Das Ziel des ersten Subtasks ist die Bestimmung von relevanten Dateneigenschaften und der zur Beurteilung dienenden Kriterien. Für die Dateneigenschaften sind zusätzlich einige Berechnungen notwendig, da nicht alle relevanten Eigenschaften direkt aus den Simulationsdaten abgelesen werden können. Die Klassifikation teilt sowohl die Dateneigenschaften als auch die Bewertungskriterien für die Schätzergebnisse in Kategorien ein. Anschließend werden für diese kategorisierten Dateneigenschaften und Bewertungskriterien Repräsentationen gesucht, so daß die Simulationsdaten in Form einer Faktenmenge dargestellt werden können.

Verteilung der Aufgaben (Task Distribution)

Da nur einer der vier Hauptaufgabenbereiche, nämlich die Kontingenztafelanalyse, vom wissensbasierten System unterstützt wird, werden die weiteren drei Bereiche, also die Datenerhebung sowie die Simulationsstudien und ihre Auswertung, hier nicht erwähnt.

Abbildung 3.8 zeigt die Verteilung der Aufgaben für die Schichtung. Die Auswahl und Entscheidungsaufgaben sind dem Benutzer des Systems zugeordnet. Allein die Ausführung der ausgewählten Methode wird durch das System geleistet. Diese Aufgabenverteilung entspricht einer konventionellen Verteilung, d.h der Benutzer hat die Kontrolle, das System ist passiv.

Ähnlich verhält es sich mit dem Subtask Homogenitätsanalyse (siehe Abb. 3.9). Die Ausführung und ein Vorschlag zur Interpretation des Ergebnisses wird vom System übernommen, die Auswahl einer Methode und die endgültige Entscheidung bleibt dem Benutzer vorbehalten.

Diese Rollenverteilung ändert sich im nächsten Subtask, der Odds Ratio Schätzung, genauer in der Durchführung einer Pilotstudie. Hier übernimmt das System die Rolle eines intelli-

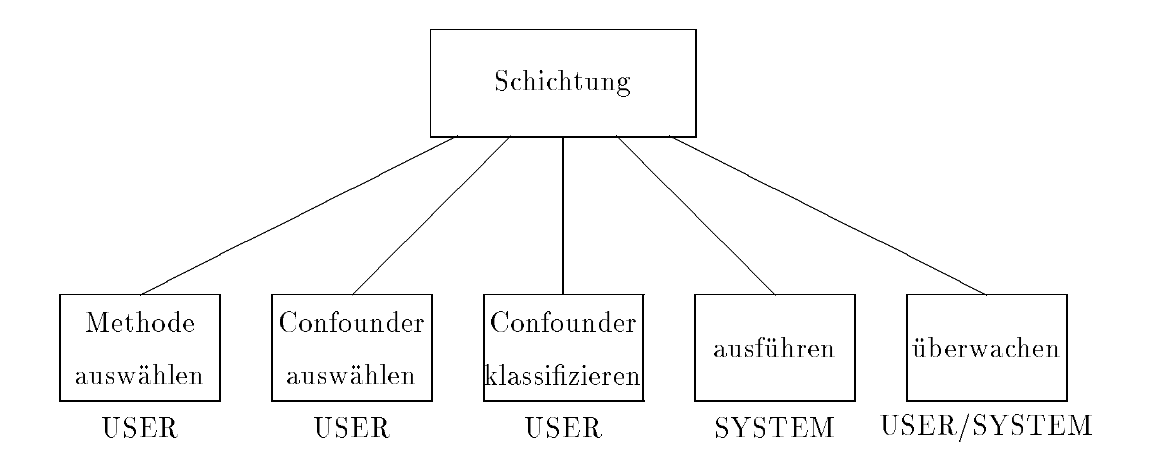


Abbildung 3.8: Verteilung der Aufgaben für die Schichtung

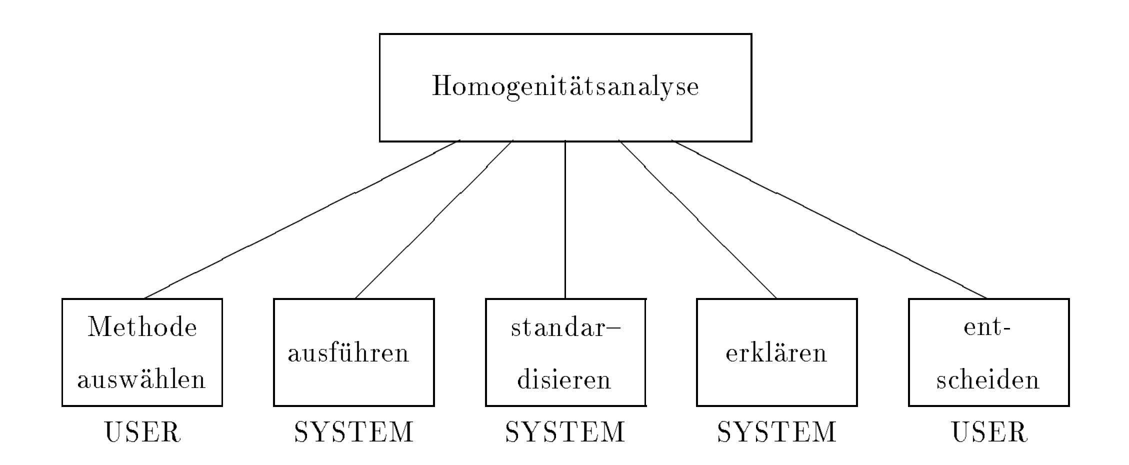


Abbildung 3.9: Verteilung der Aufgaben für die Homogenitätsanalyse

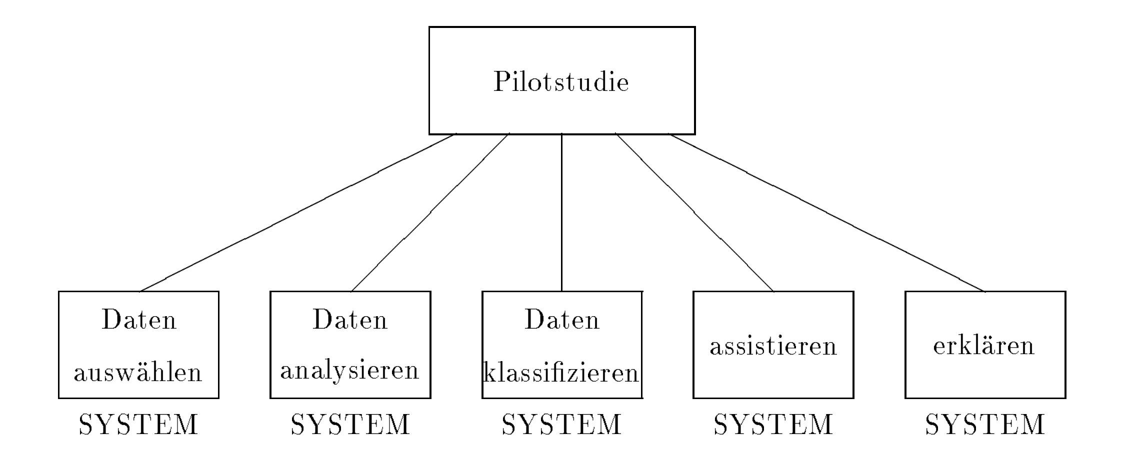


Abbildung 3.10: Verteilung der Aufgaben für die Pilotstudie

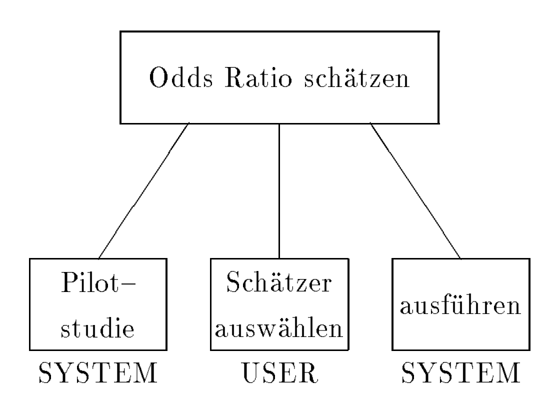


Abbildung 3.11: Verteilung der Aufgaben für die Odds Ratio Schätzung

genten, aktiven Akteurs.[6] Abbildung 3.10 zeigt die Verteilung der Aufgaben der Pilotstudie und Abbildung 3.11 die Task–Distribution für die Odds Ratio Schätzung. Die entscheidende Auswahl der Methode (des Schätzers) liegt aber auch hier letztenendes in der Hand des Benutzers.

Der Unabhängigkeitstest entspricht bzgl. der Aufgabenverteilung ungefähr der Homogenitätsanalyse. Allerdings wird dem Anwender eine konkrete Entscheidung über die Unabhängigkeit von Risikofaktor und Krankheit vom System mitgeteilt (siehe Abb. 3.12).

[6] Allerdings nur, falls dies vom Benutzer gewünscht wird.

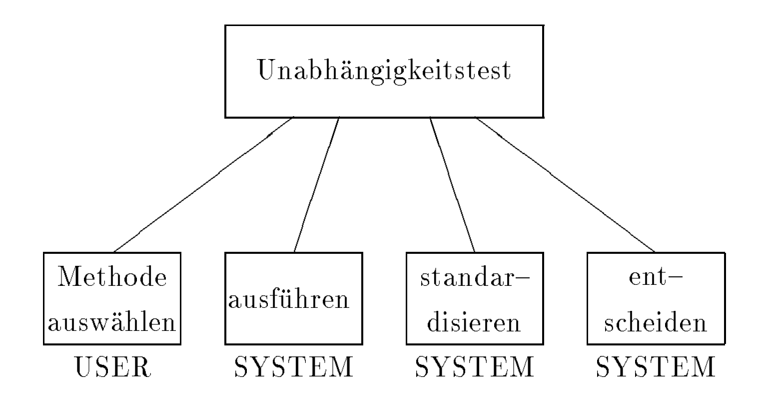


Abbildung 3.12: Verteilung der Aufgaben für Unabhängigkeitstests

3.4 Kooperationsmodell (Model of Cooperation)

Aus den Abhängigkeiten zwischen den Unteraufgaben und der Zuordnung dieser Unteraufgaben zu den Agenten Benutzer und System folgt die Kooperation, die von den Agenten zu leisten ist. Abhängigkeiten zwischen Unteraufgaben weisen auf Informationen hin, die zwischen System und Benutzer transferiert werden müssen. Diese werden auch als *Ingredients* bezeichnet. Im allgemeinen ist der Agent, der ein solches Objekt produziert auch sein Besitzer, d.h. er hat die Kontrolle über dieses Objekt. Für die Kooperation muß dann die Initiative bestimmt werden. Initiative bedeutet, daß dieser Agent die Kommunikation ausgelöst hat. Die Kooperation wird durch die Beschreibung der Transfer Tasks dargestellt.

Für die Kontingenztafelanalyse ergeben sich eine Reihe von Transfer Tasks. Ein wesentlicher Transfer Task wurde in der Beschreibung des Aufgabenwissens völlig ausgelassen. Es handelt sich dabei um die Hilfen, die vom Benutzer des Systems angefordert werden können und damit also Provide Tasks sind. Da sie allerdings generell verfügbar und in einem eigenständigen Hilfesystem vorhanden sind, werden sie nicht in der Beschreibung der einzelnen Aufgaben erwähnt.

Die meisten Erklärungen und Beratungen sind ebenfalls Provide Tasks, da sie vom Benutzer angefordert werden. Die Initiative liegt bei dem Benutzer, während das System die angeforderte Information besitzt. Auswahlen sind vorwiegend Obtain Tasks, da eine Entscheidung des Benutzers vom System angefordert wird. Gleichzeitig sind es auch Present Tasks, da die Auswahlmöglichkeiten durch das System angezeigt werden.

Die Schichtung enthält zwei Auswahl Tasks. Sie bestehen jeweils aus einem Present Task und einem Obtain Task. Die präsentierten Ingredients sind für die erste Auswahl die zur Verfügung stehenden Methoden zur Schichtung, bei der zweiten Auswahl sind es die Confounder, die dem

Datensatz zugrunde liegen. Die Ingredients der Obtain Tasks sind die ausgewählte Methode bzw. die ausgewählten Confounder. Der nächste Transfer Task ist die Klassifikation der Confounder. Es handelt sich hier um einen Obtain Task, die Ingredients sind Intervalle für die einzelnen Confounder.

Im Rahmen der Homogenitätsanalyse gibt es nur einen Transfer Task, die Erklärung des p–Werts. Sie wird vom Benutzer angefordert und ist daher ein Provide Task.

Für die Odds Ratio Schätzung kann die Pilotstudie als ein Transfer Task angesehen werden. Der Benutzer fordert die Durchführung an und erhält als Ergebnis eine Methodenempfehlung für die Fall–Kontroll–Studie. Es ist also ein Advice Task in Form eines Provide Tasks. Der zweite Transfer ist die Methodenauswahl. Hier liegt wieder ein kombinierter Present und Obtain Task vor, d.h. die zur Auswahl stehenden Schätzer werden aufgelistet.

Kapitel 4

Lernen einer Charakterisierung von Schätzern aus Simulationsstudien

Dieses Kapitel beschreibt eine Möglichkeit zur Auswertung von Simulationsstudien, die mit dem Ziel durchgeführt wurde, eine Charakterisierung von Schätzern zu lernen, speziell von Schätzern des gemeinsamen Odds Ratios in geschichteten Kontingenztafeln. Hierbei wird ein wissensbasierter Ansatz verfolgt, der auf einer Verbindung von systemunterstützter Modellierung und maschinellem Lernen beruht, d.h. das akquirierte Wissen wird modelliert und neues Wissen kann entdeckt werden.

Im Unterschied zum üblichen Knowledge Engineering basiert dieser Ansatz nicht allein auf dem Wissen der Bereichsexperten, sondern wird vorwiegend aus den in 3.3.1 beschriebenen Simulationsstudien gewonnen. Es wird eine Wissensbasis modelliert, die eine Operationalisierung der Ergebnisse der Simulationsstudien darstellt. Aus dieser Wissensbasis wird dann eine Charakterisierung der Schätzer in Form einer Menge von Regeln gelernt. Für die systemunterstützte Modellierung und das maschinelle Lernen wird das System MOBAL eingesetzt, das, wie bereits in 2.2 erwähnt, die manuelle Wissensakquisition mit dem maschinellen Lernen verbindet.

Der Modellierungsprozeß beginnt mit einer Festlegung eines Rahmens für das Modell. Darin werden relevante Dateneigenschaften von Parameterkonstellationen der Simulationsstudien bestimmt und Bewertungskriterien für die Schätzergebnisse festgelegt. Aus den Bewertungskriterien kann eine Eignung eines Schätzers in einer bestimmten Datensituation abgeleitet werden. Die Charakterisierung der Schätzer ergibt sich somit aus dem Zusammenhang zwischen der Eignung des Schätzers und den Eigenschaften der jeweiligen Datensituation. Um nicht von den konkreten Dateneigenschaften, also den genauen Werten für die Parameter, auf die Eignung zu schließen, werden diese in Kategorien eingeteilt. Das gleiche gilt für die

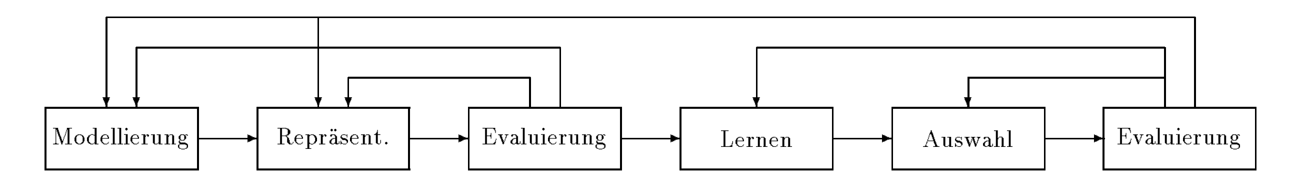


Abbildung 4.1: Phasen der Wissensakquisition

Eignung. Es sollte nicht auf konkrete Werte für die Bewertungskriterien geschlossen werden, sondern auf aussagekräftige Kategorien.
Nach der Modellierung einer Kategorieneinteilung erfolgt die Beschreibung der Repräsentation des Modells. Der Repräsentationsformalismus für die Wissensbasis ist, wie bereits in 2.2 erwähnt, eine eingeschränkte Form einer Prädikatenlogik, genauer eine funktionsfreie Hornklausellogik, die aber negierte Literale zuläßt.
Die Wissensbasis besteht aus einer Menge von Fakten, die die Simulationsstudien beschreiben. Mithilfe des in MOBAL integrierten Lernverfahrens RDT werden aus dieser Wissensbasis Regeln über die oben beschriebenen Zusammenhänge zwischen der Datenlage und dem Verhalten der Schätzer des gemeinsamen Odds Ratios gesucht. Die „besten“ Regeln aus der gelernten Regelmenge können für eine Charakterisierung der Schätzer verwendet werden.
Zusammenfassend können also folgende Phasen der Wissensakquisition festgemacht werden:

- die Modellierung der Dateneigenschaften und Bewertungskriterien,
- die Modellierung der Kategorieneinteilung,
- die Repräsentation des Modells,
- die Evaluierung des Modells,
- das Lernen einer Charakterisierung von Schätzern,
- die Auswahl einer Regelmenge und
- die Analyse der Lernergebnisse.

Es fällt auf, daß die einzelnen Phasen hier vollständig getrennt werden und der Akquisitionsprozeß damit linear statt zyklisch erscheint. Ich möchte an dieser Stelle deutlich machen, daß das nur für die *Beschreibung* der Wissensakquisition gilt, nicht für die durchgeführte Wissensakquisition selbst. Die Beziehungen zwischen den Phasen werden in Abbildung 4.1 dargestellt. Die Pfeile deuten mögliche Revisionen an.

Es sind eine ganze Reihe von Ansätzen zur Modellierung verfolgt worden, die hier nicht in ihrer chronologischen Abfolge beschrieben werden, sondern es wird auf den Ansatz fokussiert, der sich am Ende als der günstigste herausgestellt hat. Mögliche Alternativen bleiben zum Teil unerwähnt oder werden im Einzelfall nur kurz angesprochen. Wird beispielsweise in 4.2.3 von einer alternativen Modellierung der Kategorieneinteilung gesprochen, so ist zu beachten, daß diese erst nach der Repräsentation dieses Modells verworfen wurde. Die Repräsentationen für die verworfenen Modellierungen werden aus Gründen der Lesbarkeit aber nicht dargestellt.
Ebenso verhält es sich mit den unterschiedlichen Repräsentationsmöglichkeiten des Modells. Erst nach der Evaluierung des Lernvorgangs stand fest, welche Repräsentation hier geeigneter war. Dennoch werden die Lernläufe nur für eine Repräsentation beschrieben.
Das heißt, für die Beschreibung der jeweils nächsten Phase werden die „endgültigen" (abgesehen davon, daß die Modellierung generell ein unendlicher Prozeß ist) Ergebnisse der vorausgegangenen Phase zugrunde gelegt. Die Durchführung der Revisionen, die im Laufe der Wissensakquisition notwendig geworden sind, wird vollständig durch das System MOBAL unterstützt, so daß das System in allen beschriebenen Phasen eingesetzt werden konnte.
Im anschließenden Abschnitt wird zunächst das Szenario für die Modellierung beschrieben. Darin werden die grundlegenden Elemente des Sachbereichs festgelegt. Abschnitt 4.2 geht auf die Modellierung der Dateneigenschaften, der Bewertungskriterien und die Modellierung der Kategorien ein. In 4.3 wird die Repräsentation der modellierten Simulationsstudiendaten, also die Erstellung der Wissensbasis, beschrieben. In Abschnitt 4.4 erfolgt eine Evaluierung des Modells. In den folgenden Abschnitten werden die Lernaufgabe und die Lernläufe beschrieben. In 4.7 werden einige Regeln, die durch eine frühere Auswertung von Simulationsstudien 'per Hand' gewonnen worden sind, auf Konsistenz mit der erstellten Wissensbasis überprüft. Daran anschließend wird eine Regelmenge aus den gelernten Regeln ausgewählt und anhand der festgelegten Bewertungskriterien analysiert. Bei der ausgewählten Regelmenge handelt es sich dann um die Regeln, die in das wissensbasierte System CORA integriert werden. Das Einbinden dieser Regeln wird in 4.10 erläutert.

4.1 Szenario

In diesem Abschnitt wird ein Rahmen für das zu bildende Modell (Szenario) abgesteckt. Darin werden die grundlegenden Elemente des Sachbereichs festgehalten. Die wesentlichen Konzepte (siehe Abbildung 4.2) sind hier

- die Simulationsstudie,
- die Parameterkonstellationen in der Simulationsstudie,

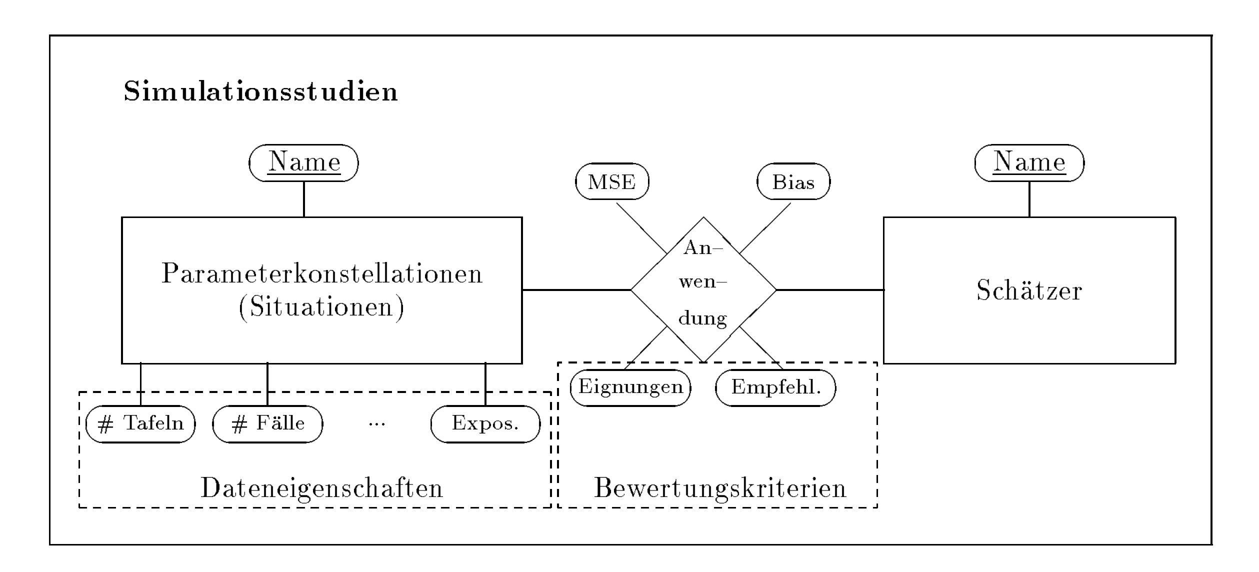


Abbildung 4.2: Entity Relationship Modell des Sachbereichs

- die Schätzer des gemeinsamen Odds Ratios,
- die Dateneigenschaften der Parameterkonstellationen und
- die Bewertungskriterien für die Anwendung eines Schätzers in einer Parameterkonstellation (die Schätzung des gemeinsamen Odds Ratios).

Die dieser Modellierung zugrundeliegende Simulationsstudie ist bereits in 3.3.1 beschrieben worden. Sie umfaßt 240 unterschiedliche Parameterkonstellationen (Situationen), in denen folgende sechs Punktschätzer des gemeinsamen Odds Ratios untersucht worden sind[1]:

- der Mantel Haenszel–,
- der Woolf–,
- der Breslow Liang–,
- der Woolf Jackknife–,
- der Jackknife II– und
- der modifizierte Jackknife II–Schätzer.

Die Dateneigenschaften der Simulationsdaten entsprechen den Eigenschaften, durch die auch die Datenlage einer Fall–Kontroll–Studie beschrieben werden kann. Es können folgende relevante Dateneigenschaften festgemacht werden:

[1] Die Schätzer werden, wie bereits erwähnt, mit MH, W, BL, W_JK, JK und JK_ii abgekürzt.

- die Anzahl der Tafeln,
- das gemeinsame Odds Ratio,
- die Anzahl der Fälle und Kontrollen und
- die Expositionswahrscheinlichkeiten.

Durch diese Parameter werden die Situationen der Simulationsstudien beschrieben. Auch weitere Dateneigenschaften können aus diesen 'grundlegenden' Parametern abgeleitet werden. Die Bestimmung dieser abgeleiteten Dateneigenschaften gehört jedoch schon zur Modellierung des Sachbereichs und wird daher in 4.2 näher beschrieben.
Von den Dateneigenschaften soll auf die Eignung der Schätzer geschlossen werden. Um eine Eignung eines Schätzers beurteilen zu können, werden folgende Bewertungskriterien herangezogen:

- der Bias, also die Verzerrung, und
- der MSE (Mean Squared Error), also der mittlere quadratische Fehler.

Ein weiteres Kriterium, das in der Simulationsstudie ebenfalls angegeben wurde, ist die Standardabweichung, die aber, wie in 3.3.1 schon erwähnt, aus dem Bias und dem MSE berechnet werden kann und daher nicht mehr zusätzlich zur Beurteilung herangezogen wird.
Die hier aufgeführten Dateneigenschaften und Bewertungskriterien können direkt aus der Dokumentation der Simulation abgelesen werden.

4.2 Modellierung

Dieser Abschnitt umfaßt die Modellierung des Sachbereichswissen, dessen Rahmen im Szenario festgelegt worden ist. Dazu erfolgt zunächst die Modellierung der Dateneigenschaften. Mit den in 4.1 beschriebenen Eigenschaften können zwar alle Parameterkonstellationen vollständig beschrieben werden; es können darüber hinaus aber noch eine Reihe weiterer Eigenschaften bestimmt werden, die sich aus den schon beschriebenen ableiten lassen. Die Berechnung dieser zusätzlichen Eigenschaften wird als sinnvoll erachtet, da so die Charakterisierung der Schätzer, die auf diesen Dateneigenschaften basiert, an Ausdrucksfähigkeit gewinnt.
Anschließend werden die Kriterien modelliert, die zur Bewertung der Schätzergebnisse herangezogen werden. Dazu wird die getrennte Bewertung durch den Bias und den MSE, die im Szenario angedeutet wurde, zu einer Gesamteignung jedes Schätzers und zu einer Empfehlung eines Schätzers in einer Situation zusammengefaßt.

Abschließend erfolgt eine Modellierung von Kategorien, in die die Dateneigenschaften und die Bewertungskriterien eingeteilt werden. Sowohl die Eigenschaften als auch die Bewertungen sind numerisch. Die Kategorien bestehen daher aus Intervallen, die die konkreten Werte für die Eigenschaften und Bewertungen zusammenfassen und so von diesen Werten abstrahieren.

4.2.1 Modellierung der Dateneigenschaften

Aus den im Szenario beschriebenen Parametern können eine Reihe weiterer Dateneigenschaften abgeleitet werden. Eine als wichtiger Einflußfaktor auf das Verhalten der Schätzer angesehene Dateneigenschaft ist das Verhältnis von Kontrollen und Fällen. Desweiteren ist auch die Differenz des maximalen Werts der Expositionswahrscheinlichkeiten und des minimalen Werts von besonderem Interesse. Beide Eigenschaften können auf einfache Weise aus den Grundeigenschaften berechnet werden.

Zu beachten ist, daß es sich bei den Eigenschaften Fallanzahl, Kontrollenanzahl, Verhältnis von Kontrollen und Fällen und den Expositionswahrscheinlichtkeiten um Eigenschaften der einzelnen Schichten handelt. Besteht ein Datensatz aus zehn Schichten, gibt es damit zehn unterschiedliche Werte für die Fallanzahl, die Anzahl der Kontrollen etc.

Die Dateneigenschaften können also in zwei Gruppen eingeteilt werden:

1. Eigenschaften für die gesamte Parameterkonstellation und

2. Eigenschaften, die sich auf die Schichten einer Situation beziehen.

Die Beschreibung einer Parameterkonstellation ist für die zweite Gruppe von Eigenschaften sehr komplex. Zur Komprimierung können Durchschnittswerte für diese Eigenschaften aus den Werten für die Schichten in einer Konstellation berechnet werden.

Da bei der Berechnung der Durchschnittswerte Informationen verloren gehen, wird die Betrachtung der Durchschnittswerte alleine als nicht ausreichend erachtet. Es werden neue Dateneigenschaften eingeführt, die mithilfe des sogenannten Gini–Koeffizienten aus den ursprünglichen Eigenschaften abgeleitet werden können. Durch diesen Koeffizienten kann die Balanciertheit der Eigenschaften über die Schichten beurteilt werden, genauer: man bewertet die Aufteilung in die Kategorien für diese Eigenschaften, die in Abschnitt 4.2.3 beschrieben wird. Zur Verdeutlichung dieses Balanciertheitsmaßes betrachte man folgendes Beispiel:

> Es liegt eine Parameterkonstellation mit zwei Schichten vor. Die erste Schicht umfaßt 20 Fälle, die zweite Schicht 30 Fälle. Angenommen, bezüglich der Fallanzahlen werden nun zwei Kategorien gebildet, nämlich eine Kategorie für Anzahlen bis zu 25 Fällen und eine Kategorie für größere Anzahlen. Die erste Schicht der betrachteten Parameterkonstellation fällt in die erste Kategorie, die zweite Schicht

wird dagegen in die Kategorie für größere Anzahlen eingeordnet. Damit liegt in dieser Situation eine balancierte Aufteilung der Fallanzahlen vor, da die Schichten gleichmäßig in die vorhandenen Kategorien fallen. Betrachtet man aber eine Konstellation in der beide Schichten 20 Fälle umfassen, so wird die zweite Kategorie nicht verwendet, weil beide Schichten in die Kategorie für kleine Fallanzahlen eingeordnet werden. Es handelt sich somit um eine unbalancierte Aufteilung der Fallanzahlen in die Kategorien.

Der Gini–Koeffizient, hier abgekürzt durch GK, wird folgendermaßen berechnet:

$$GK := \frac{K}{K-1}(\sum_{i=1}^{B}(K_{i-1} + K_i)(l_i - 1)),$$

wobei $K_i = \frac{\sum_{j=1}^{i} h_j}{\sum_{j=1}^{B} h_j}$ und $l_i = \frac{S_i}{\sum_{j=1}^{B} S_j}$;

K: Anzahl der Schichten,

S_i: Summe der Ausprägungen der Eigenschaft in der Kategorie i,
z.B. die Summe der Fallanzahlen für die Kategorie „viele Fälle“,

H_i: Anzahl der Schichten in der i–ten Kategorie und $h_i = \frac{H_i}{K}$,

B: Anzahl der Kategorien.

Die Werte für den Gini–Koeffizienten liegen zwischen 0 und 1. Ist der Wert des Koeffizienten 0, so handelt es sich um eine vollständig gleichmäßige Aufteilung in die vorhandenen Kategorien. Wenn der Gini–Koeffizient den Wert 1 erreicht, ist die Aufteilung sehr unbalanciert.
Damit können drei neue Dateneigenschaften für die Fall-Kontroll-Studien abgeleitet werden, nämlich:

- die Balanciertheit der Fallanzahlen in einer Parameterkonstellation,
- die Balanciertheit der Größe des Verhältnisses der Kontrollen und Fälle und
- die Balanciertheit der Lage der Expositionswahrscheinlichkeiten.

Durch die ersten beiden Eigenschaften wird indirekt auch die Balanciertheit der Anzahl der Kontrollen ausgedrückt.

4.2.2 Modellierung der Bewertungskriterien

Ausgehend von den zwei grundlegenden Bewertungskriterien, dem Bias und dem MSE, können, ähnlich wie bei der Modellierung der Dateneigenschaften, weitere Kriterien zur Beurteilung der Schätzergebnisse abgeleitet werden. Es kann eine Rangfolge der Schätzer gebildet

werden und die beiden Einzelkriterien können zu einer Gesamteignung eines Schätzers in einer Situation zusammengefaßt werden. Die Berechnung dieser Gesamteignung hängt mit der Kategorieneinteilung von Bias und MSE zusammen und wird daher erst nach der Modellierung der Kategorien in 4.2.3 detailliert erläutert.

Neben der Beurteilung der Schätzer durch die absoluten Werte für Bias und MSE können die relativen Werte für diese Kriterien eingeführt werden. Die Bewertung anhand dieser Kriterien erwies sich aber als ungünstig und wurde daher später wieder verworfen. Die Motivation für ihre Einführung bestand darin, daß es vom statistischen Gesichtspunkt häufig als problematisch angesehen wird, nur den absoluten Bias und MSE zu betrachten. In den Simulationsdaten gibt es große Unterschiede bzgl. der absoluten Bewertungskriterien zwischen den Parameterkonstellationen. Diese sind auf die unterschiedliche Größe des gemeinsamen Odds Ratios zurückzuführen, d.h., daß bei einem großen zu schätzenden Odds Ratio der absolute Bias und MSE in der Regel größer sind als bei kleinen zu schätzenden Werten. Die Beurteilung der Schätzergebnisse durch die relativen Kriterien liefert aber eine zu geringe Trennung der Schätzer.

Aber auch bereits die absoluten Werte für die Kriterien Bias und MSE bieten oft eine schlechte Differenzierung der Schätzer. Die Werte für die einzelnen Kriterien liegen bei den Schätzern innerhalb einer Parameterkonstellation fast immer sehr nah zusammen. Aber besonders beim MSE gibt es zwischen den Parameterkonstellationen große Unterschiede. Eine einheitliche Beurteilung fällt daher schwer. Aus diesem Grund wurde es als sinnvoll erachtet, Reihenfolgen der Schätzer bezüglich der Kriterien Bias und MSE zu bilden, die zusätzlich zur Beurteilung herangezogen werden. Die beiden Rangfolgen für den Bias und den MSE können dann zusammengefaßt werden. Sind dabei mehrere Schätzer gleich, soll der bessere Bias den Ausschlag geben[2]. Dieses Vorgehen wurde gewählt, da anhand des Bias eine bessere Trennung zwischen den Schätzern vorgenommen werden konnte. Mit der entstandenen Gesamtrangfolge wird der beste Schätzer in dieser Situation ermittelt.

4.2.3 Modellierung der Kategorien

Um aussagekräftige Regeln lernen zu können, müssen Kategorien gefunden werden, in die die Eigenschaften eingeteilt werden können. Bislang bestehen sowohl die Dateneigenschaften als auch die Bewertung der Eignung im Modell jeweils aus konkreten Zahlenwerten. Die Charakterisierung der Schätzer durch die zu lernenden Regeln sollte aber nicht auf diese Werte zurückgreifen, da solche Regeln, wie etwa die folgende, wenig Aussagekraft besitzen:

[2] Wenn z.B. ein Schätzer bzgl. Bias der beste und MSE der zweitbeste war, ein anderer Schätzer bzgl. MSE der beste und Bias der zweitbeste wäre der erste Schätzer also dem zweiten vorzuziehen.

> Hat eine Situation ein gemeinsames Odds Ratio von 3.5 und gibt es 30 Fälle, so hat Schätzer Z einen Bias von 0.034.

Es ist sinnvoll, von diesen Werten zu abstrahieren, indem sie in größere Intervalle, die Kategorien, zusammengefaßt werden. Die „richtige" Wahl dieser Kategorien ist entscheidend für das spätere Lernen (siehe 4.9). Der Prozeß der Kategorieneinteilung ist nicht nur aus diesem Grund ein besonders schwieriger interaktiver Prozeß zwischen Wissensingenieur und Experte. Bei der Kategorieneinteilung ist zu beachten, daß sehr große Kategorien dazu führen, daß die Zusammenhänge zwischen den Dateneigenschaften und dem Verhalten der Schätzer nicht mehr gelernt werden können (siehe dazu auch 4.9). Kleine Kategorien erhöhen dagegen die ohnehin sehr große Komplexität des Modells. Als dritter Punkt ist auch das Simulationsdesign zu beachten. Es müssen ausreichend Parameterkonstellationen in allen Kategorien vorhanden sein, damit die später zu lernenden Regeln auf einer aussagekräftigen Grundlage aufbauen können. Wichtig ist, daß die Kategorieneinteilung für weitere Simulationsstudien aufrecht zu halten ist. So sollte die Kategorieneinteilung ermöglichen, daß Studiendaten sowohl aus dem Large–Strata Modell als auch aus dem Sparse–Data Modell in die gebildeten Kategorien eingeteilt werden können. Da hier beispielsweise nur eine Studie, der das Large–Strata Modell zugrunde liegt, betrachtet wurde, gibt es u.a. keine Situationen, die eine sehr große Anzahl von Tafeln besitzen. Dennoch wurde eine Kategorie hierfür gebildet, da eine spätere Einbeziehung von Studien aus dem Sparse–Data Modell ohne Änderungen der Kategorieneinteilungen möglich sein sollte.

Dieser Prozeß der Kategorienbildung soll hier nicht dargestellt werden. Es wird im folgenden lediglich die endgültig gewählte Einteilung vorgestellt, siehe Tabelle 4.1. Diese Tabelle gibt die Einteilung für die Dateneigenschaften und die Bewertungskriterien wieder. Für die Durchschnittswerte gilt die gleiche Einteilung wie für die Werte in den einzelnen Schichten, d.h. sowohl die Anzahl der Fälle, der Kontrollen und ihr Verhältnis und die Expositionswahrscheinlichkeiten in den einzelnen Schichten als auch die durchschnittlichen Werte hierfür werden gemäß Tabelle 4.1 bewertet. Für die Kategorienbildung der Werte des Gini–Koeffizienten gibt es eine Einteilung, die für alle drei oben genannten Balanciertheitseigenschaften einheitlich ist.

Für einige Parameter war die Kategorienbildung unproblematisch und hat sich im Laufe der Modellierung nicht oder nur kaum geändert. Dazu gehören z.B. die Kategorien für das gemeinsame Odds Ratio. Die bestehende Einteilung wird bereits bei der Beschreibung des Designs der Simulationsstudie durch die Expertin motiviert (siehe 3.3.1).

Andere Kategorien, insbesondere die für den Bias und den MSE, wurden häufig geändert. Für diese Parameter ist es wichtig, eine möglichst gute Trennung der Schätzer zu erhalten. Daher wurde bei diesen Parametern weniger auf eine „inhaltlich" motivierte Einteilung Wert gelegt,

im Gegensatz zu der Kategorisierung der Dateneigenschaften, sondern vorwiegend untersucht, mit welchen Werten für die Kategoriegrenzen die gewünschte Aufteilung der Schätzer durchgeführt werden kann. So kommt man dann zu einer Einteilung, die einen Bias von 0.05 bereits als einen mittleren Bias klassifiziert. „Inhaltlich“ gesehen ist ein solcher Bias sicher noch als sehr klein zu bezeichnen. Diese strenge Einteilung wirkt sich natürlich auf die daraus entstehende Gesamteignung und auch auf die Einteilung der Empfehlungen aus. Daher ist es wichtig, später bei der Verwendung der gelernten Charakterisierung deutlich zu machen, daß eine sehr strenge Einteilung zugrunde gelegen hat.
Nachdem nun die Kategorien für die Bewertungskriterien Bias und MSE gebildet worden sind, kann die Modellierung der kategorisierten Eignungen für die Schätzer erfolgen. Hierfür können die Einzelbewertungen zu einer Gesamteignung verknüpft werden. Für alle Schätzer werden vier Stufen von Eignungen gebildet. Die Stufen unterscheiden sich darin, wie gut die Werte für den Bias und den MSE für den Schätzer in der Situation waren. In der ersten Stufe sind die Schätzer, die in Kategorien für einen sehr kleinen Bias und einen sehr kleinen MSE eingeteilt wurden. Um in die zweite Kategorie, also eine gute Eignung eingeordnet zu werden, gibt es drei Möglichkeiten:

1. einer der Werte für Bias oder MSE ist sehr klein und der andere Wert ist klein,
2. einer der Werte ist sehr klein und der andere Wert ist mittelgroß oder,
3. beide Werte sind klein.

Vier mögliche Kombinationen führen zu einer Einteilung in die dritte Kategorie:

1. einer der Werte für Bias oder MSE ist klein und der andere ist mittelgroß,
2. einer der Werte für Bias oder MSE ist klein und der andere ist groß,
3. einer der Werte ist sehr klein und der andere ist groß oder,
4. beide Werte sind mittelgroß.

Die Tabelle 4.2 faßt die Stufenbildung für die Eignung zusammen. Dabei bedeutet (X/Y), daß der Bias aus Kategorie X stammen kann und der MSE aus Kategorie Y und umgekehrt, also daß der MSE aus Kategorie X und der Bias aus Kategorie Y stammen kann. Eine 1 steht dabei für die Kategorie mit sehr kleinen Werten, eine 2 für die Kategorie mit kleinen Werten usw.
Auch die Empfehlungen der Schätzer werden bewertet, d.h. entsprechend der Eignung des empfohlenen Schätzers wird die Empfehlung eingeteilt in eine sehr gute Empfehlung, eine gute Empfehlung, eine mittere Empfehlung und eine schlechte Empfehlung.

Schichtanzahl	**S**	**Odds Ratio**	**Ψ**
wenige Schichten	$S < 5$	Odds Ratio = 1	$\Psi = 1$
mittlere Anzahl an Schichten	$5 \leq S < 10$	kleines Odds Ratio	$1 < \Psi \leq 2$
viele Schichten	$10 \leq S < 50$	mittleres Odds Ratio	$2 < \Psi \leq 7$
sehr viele Schichten	$50 \leq S$	großes Odds Ratio	$7 < \Psi$
Fallanzahl	**F**	**Kontrollenanzahl**	**K**
wenige Fälle	$F \leq 5$	wenige Kontrollen	$K \leq 5$
mittel viele Fälle	$5 < F \leq 20$	mittel viele Kontrollen	$5 < K \leq 20$
viele Fälle	$20 < F \leq 100$	viele Kontrollen	$20 < K \leq 100$
sehr viele Fälle	$100 < F$	sehr viele Kontrollen	$100 < K$
Verhältnis K/F	**V**	**Balanciertheit**	**GK**
balanciert	$V \leq 1.25$	balanciert	$GK = 0$
mittel balanciert	$1.25 < V \leq 3$	mittelbalanciert	$1 < GK \leq 0.5$
unbalanciert	$3 < V$	unbalanciert	$0.5 < GK \leq 1$
Lage der Expo–sitionswahrschein–lichkeiten	**E**	**Differenzen der Expositionswahr–scheinlichkeiten**	**D**
unterer Rand	$E \leq 0.3$	kleine Differenz	$D \leq 0.2$
zentriert	$0.3 < E \leq 0.7$	große Differenz	$0.2 < D$
oberer Rand	$0.7 < E$		
Bias	**B**	**MSE**	**M**
sehr kleiner Bias	$B < 0.005$	sehr kleiner MSE	$M < 0.01$
kleiner Bias	$0.005 \leq B < 0.05$	kleiner MSE	$0.01 \leq M < 0.1$
mittlerer Bias	$0.05 \leq B < 0.5$	mittlerer MSE	$0.1 \leq M < 1$
großer Bias	$0.5 \leq B$	großer MSE	$1 \leq M$

Tabelle 4.1: Kategorienbildung

sehr gute Eignung	(1/1)
gute Eignung	(1/2)(1/3)(2/2)
mittlere Eignung	(2/3)(2/4)(1/4)(3/3)
schlechte Eignung	(3/4)(4/4)

Tabelle 4.2: Stufenbildung für die Eignung

In Abhängigkeit davon, aus welcher Stufe die Empfehlungen stammen, kann der Benutzer der zu lernenden Regeln später darauf hingewiesen werden, ob in dieser Situation eher mit guten oder schlechten Ergebnissen zu rechnen ist.

4.3 Repräsentationen der Modellierung

Der Abschnitt Repräsentationen der Modellierung beschreibt den Aufbau einer Wissensbasis. Wie schon zu Anfang des Kapitels erwähnt wurde, sind die Phasen hier getrennt dargestellt. Es sollte beachtet werden, daß diese Phase, die Repräsentation, mehrmals durchlaufen wurde. Hier werden aber jeweils nur Repräsentationen für die endgültig ausgewählte Modellierung der Dateneigenschaften und Bewertungskriterien vorgestellt. Desweiteren liegt der Schwerpunkt auf der später gewählten Repräsentation. Es wird jeweils nur eine der möglichen Alternativen kurz erwähnt.
Die beiden ersten Schritte beim Aufbau einer Wissensbasis bestehen aus

1. der Repräsentation der Dateneigenschaften und

2. der Repräsentation der Schätzergebnisse.

Hier werden die wohl wichtigsten Entscheidungen für die Repräsentation getroffen. Die Objekte, Relationen und Eigenschaften können auf unterschiedliche Weisen im Modell repräsentiert werden. Sie können sowohl in Form von Prädikaten als auch von Sorten, d.h. der Argumente der Prädikate, in die Wissensbasis eingehen.
Nach Durchführung dieser Schritte besteht die Wissensbasis aus Fakten, die die konkreten Werte für die Eigenschaften und Ergebnisse repräsentieren. Für die Umsetzung der abgeleiteten Dateneigenschaften müssen einfache Regeln in die Wissensbasis eingegeben werden. Nach der Repräsentation der Werte aus den Simulationsstudien sind in einem dritten Schritt auch die gebildeten Kategorien zu repräsentieren. Dazu ist die Faktenbasis wiederum durch die Eingabe entsprechender Regeln zu erweitern. Liegen schließlich die kategorisierten Schätzergebnisse in der Wissensbasis vor, so kann die Umsetzung der Modellierung der Eignungen durch entsprechende Regeln erfolgen. Für die Empfehlung muß die zu bildende Rangfolge ebenfalls durch die Eingabe einer Regelmenge umgesetzt werden. Die Repräsentation der Eignungen und Empfehlungen ist komplexer als die Ableitung neuer Dateneigenschaften oder die Kategorienbildung. Die entstandene Regelmenge wirkt auf den ersten Blick etwas undurchsichtig und umständlich. Sie führt dazu, daß eine ganze Reihe von Fakten abgeleitet werden, die aber, nachdem man die gewünschten Ergebnisse gewonnen hat, nicht mehr benötigt werden.
In den folgenden Unterabschnitten werden nun die gerade festgelegten Schritte für den Aufbau der Wissensbasis beschrieben.

4.3.1 Repräsentationen der Dateneigenschaften

Für die Repräsentation der Dateneigenschaften werden zwei Alternativen vorgestellt, von denen die zuerst beschriebene später weiterverwendet wird. Die zweite Repräsentation wird lediglich angegeben, um zu verdeutlichen, daß jeweils mehrere Möglichkeiten bestehen. Der Leser sollte dadurch eine Vorstellung davon bekommen, welche Entscheidungen hier getroffen werden müssen.

In der ersten Repräsentation gibt es für jede schichtunabhängige Dateneigenschaft ein zweistelliges Prädikat. An der ersten Stelle steht die Situationsnummer. Sie ist ein eindeutiger Bezeichner für die Parameterkonstellation aus der Simulationsstudie. An der zweiten Stelle im Prädikat stehen die Ausprägungen der Eigenschaften, also die Werte des Odds Ratios, die Anzahl der Schichten oder die Differenz der Expositionswahrscheinlichkeiten. Es ist sinnvoll, die Prädikatnamen so zu wählen, daß sie auf die jeweils repräsentierte Eigenschaft deuten. Folgende Prädikate werden konstruiert:

$\mathit{oddsratio}(< \mathit{situationsnummer} >, < \mathit{groesse} >)$
$\mathit{schichtanzahl}(< \mathit{situationsnummer} >, < \mathit{anzahl} >)$
$\mathit{differenz_expo}(< \mathit{situationsnummer} >, < \mathit{differenz} >)$

Ebenso werden die durch den Gini–Koeffizienten berechneten Werte für die Balanciertheit der schichtabhängigen Eigenschaften repräsentiert. Hier entehen die drei Prädikate:

$\mathit{balanciertheit_faelle}(< \mathit{situationsnummer} >, < \mathit{gk} >)$
$\mathit{balanciertheit_verhaeltnis}(< \mathit{situationsnummer} >, < \mathit{gk} >)$
$\mathit{balanciertheit_expo}(< \mathit{situationsnummer} >, < \mathit{gk} >)$

Für die weiteren Eigenschaften gilt, daß sie in den einzelnen Schichten einer Situation unterschiedlich sind. In die Prädikate für diese Eigenschaften muß daher ein drittes Argument eingefügt werden, in dem die Schichtnummer angegeben wird. Sie zeigt an, zu welcher Schicht der Situation die Ausprägungen dieser Eigenschaften gehören. Es ergeben sich damit die Prädikate:

$\mathit{fallanzahl}(< \mathit{situationsnummer} >, < \mathit{schichtnummer} >, < \mathit{anzahl} >)$
$\mathit{kontrollenanzahl}(< \mathit{situationsnummer} >, < \mathit{schichtnummer} >, < \mathit{anzahl} >)$
$\mathit{exposition}(< \mathit{situationsnummer} >, < \mathit{schichtnummer} >, < \mathit{wert} >)$
$\mathit{verh_k/f}(< \mathit{situationsnummer} >, < \mathit{schichtnummer} >, < \mathit{wert} >)$

Sollen für diese vier Eigenschaften die Durchschnittswerte angegeben werden, so bietet sich die Repräsentation durch folgende Prädikate an:

$durchschnitt_fallanzahl(< situationsnummer >, < anzahl >)$
$durchschnitt_kontrollenanzahl(< situationsnummer >, < anzahl >)$
$durchschnitt_exposition(< situationsnummer >, < wert >)$
$durchschnitt_verh_k/f(< situationsnummer >, < wert >)$

Einige Werte für die Dateneigenschaften können nicht direkt aus der Dokumentation der Simulationsdaten übernommen werden. Es handelt sich hierbei um die abgeleiteten Eigenschaften, die in 4.2.1 beschrieben wurden, also das Verhältnis von Kontrollen und Fällen, die Differenz der Expositionswahrscheinlichkeiten und die Durchschnittswerte. Diese Werte können durch die Eingabe einfacher Regeln berechnet werden, wie beispielsweise durch die Regel:

```
fallanzahl(Sit, S, F) & kontrollenanzahl(Sit, S, K) & div(K, F,
VKF)
→ verh_k/f(Sit, S, VKF)
```

Hiermit wird das Verhältnis „VKF“ von Kontrollen und Fällen in der Schicht „S“ der Situation „Sit“ berechnet. In diesen Regeln wird auf Built–In Prädikate zurückgegriffen (hier z.B. `div`), die in 2.2 beschrieben worden sind. Die Berechnung der Gini–Koeffizienten ist auf diese Weise nicht zu realisieren und wird daher extern durchgeführt.
Durch die Einführung der oben aufgeführten Prädikate kann eine Wissensbasis aufgebaut werden, die die Dateneigenschaften der Parameterkonstellationen aus der Simulationsstudie repräsentiert. Das folgende Beispiel stellt eine Wissensbasis dar, die die erste Situation aus der Simulationsstudie bzgl. der Dateneigenschaften beschreibt:

```
oddsratio(sit_1,1.0)
schichtanzahl(sit_1,2)
differenz_expo(sit_1,0.1)
balanciertheit_faelle(sit_1,1)
balanciertheit_verhaeltnis(sit_1,0)
balanciertheit_expo(sit_1,1)
fallanzahl(sit_1,1,20)
```

```
fallanzahl(sit_1,2,30)
kontrollenanzahl(sit_1,1,60)
kontrollenanzahl(sit_1,2,90)
exposition(sit_1,1,0.2)
exposition(sit_1,2,0.3)
verh_k/f(sit_1,1,3)
verh_k/f(sit_1,2,3)
durchschnitt_fallanzahl(sit_1,25)
durchschnitt_kontrollenanzahl(sit_1,75)
durchschnitt_exposition(sit_1,0.25)
durchschnitt_verh_k/f(sit_1,3)
```

Aus diesen Fakten kann beispielsweise abgelesen werden, daß das Odds Ratio in dieser Situation bei 1.0 liegt und zur ersten Schicht 20 Fälle gehören.
Um alle Situationen zu charakterisieren, sind für die Prädikate, die unabhängig von den Schichten sind, jeweils 240 Fakten notwendig, für die weiteren Prädikate 1360, da für jede Schicht einer Situation ein Faktum benötigt wird[3]. Insgesamt umfaßt die Wissensbasis damit (10*240)+(4*1360)=7840 Fakten zur Beschreibung der Dateneigenschaften.

Es sind jedoch weitere Möglichkeiten zur Repräsentation vorhanden, so z.B. die folgende Alternative. Es werden alle Ausprägungen der schichtunabhängigen Eigenschaften als Argumente in einem Prädikat zusammengefaßt. Für die Eigenschaften, die in den Schichten unterschiedlich sind, wird ein zweites Prädikat konstruiert. Es ergibt sich die folgende Spezifikation für die beiden Prädikate:

$$parameterkonstellation(< situationsnummer >, < oddsratio >, < tafelanzahl >,$$
$$< differenzderexpos >, < balanciertheit_faelle >, < balanciertheit_verhaeltnis >,$$
$$< balanciertheit_expo >, < durchschnitt_fallanzahl >,$$
$$< durchschnitt_kontrollenanzahl >, < durchschnitt_exposition >,$$
$$< durchschnitt_verh_k/f >)$$

$$konstellation_in_schicht(< situationsnummer >, < schichtnummer >, < fallanzahl >,$$
$$< kontrollenanzahl >, < exposition >, < verh_k/f >)$$

Das Beispiel zeigt die Dateneigenschaften der ersten Situation in dieser Repräsentation:

[3] Es gibt 1360 Schichten in der repräsentierten Simulationsstudie.

```
parameterkonstellation(sit_1,1.0,2,0.1,1,0,1,25,75,2.5,3)
konstellation_in_schicht(sit_1,1,20,60,0.2,3)
konstellation_in_schicht(sit_1,2,30,90,0.3,3)
```

Um alle Situationen in dieser Repräsentation zu charakterisieren, sind 240 Fakten für das erste Prädikat und 1360 für das zweite notwendig, d.h. 1600 Fakten, also wesentlich weniger als in der ersten Repräsentationsvariante. Allerdings sind die Fakten „unübersichtlicher“ geworden, da sich die Stelligkeit der Prädikate erhöht hat. Aus der im Beispiel vorgestellten Wissensbasis läßt sich schwerer ablesen, daß das Odds Ratio 1.0 beträgt und es in der ersten Schicht 20 Fälle gibt.

4.3.2 Repräsentationen der Schätzergebnisse

Die Beschreibung der Repräsentation der Schätzergebnisse stellt ähnlich wie im vorangegangenen Abschnitt zwei Alternativen vor, von denen die erste weiter verfolgt wird. Wie schon erwähnt, erfolgt hier nur der erste Teil der Umsetzung der Schätzergebnisse, also nur die Repräsentation der einfachen Bewertungskriterien Bias und MSE.
In der ersten Repräsentationsvariante gibt es für diese beiden Bewertungskriterien jeweils ein Prädikat, das als Argumente die Situationsnummer, den zugehörigen Namen des Schätzers und den Wert des Kriteriums enthält, nämlich die Prädikate:

$bias(< situationsnummer >, < schaetzer >, < wert >)$
$mse(< situationsnummer >, < schaetzer >, < wert >)$

Eine weitere Möglichkeit zur Repräsentation der Schätzergebnisse besteht darin, sechs Prädikate für die unterschiedlichen Schätzer zu bilden und die Werte für den Bias und den MSE in diesem Prädikat zusammenzufassen. Damit hat man die Prädikate:

$mantel_haenszel(< situationsnummer >, < wert_fuer_bias >, < wert_fuer_mse >)$
$woolf(< situationsnummer >, < wert_fuer_bias >, < wert_fuer_mse >)$
$breslow_liang(< situationsnummer >, < wert_fuer_bias >, < wert_fuer_mse >)$
$jackknife_typ_I(< situationsnummer >, < wert_fuer_bias >, < wert_fuer_mse >)$
$jackknife_typ_II(< situationsnummer >, < wert_fuer_bias >, < wert_fuer_mse >)$
$woolf_jackknife(< situationsnummer >, < wert_fuer_bias >, < wert_fuer_mse >)$

In der ersten Repräsentation entstehen 1920 Fakten, in der zweiten 960, wenn die Ergebnisse für alle 240 Situationen repräsentiert werden. Da der Bias und der MSE durch die Kategorienbildung getrennt bewertet werden, scheint die erste Repräsentation günstiger zu sein.

4.3.3 Repräsentationen der Kategorien für die Dateneigenschaften und Schätzergebnisse

Der folgende Unterabschnitt beschäftigt sich damit, eine geeignete Darstellung für die durch die Kategorienbildung eingeteilten Eigenschaften und Bewertungen zu finden. Dafür müssen zunächst Prädikate für die in Kategorien eingeteilten Eigenschaften und Bewertungen konstruiert werden, um dann Regeln eingeben zu können, die die neuen Fakten ableiten. Wieder werden zwei alternative Wege aufgezeigt.
Die erste Repräsentationsmöglichkeit läßt die Bezeichnungen für die Kategorien mit in den Prädikatnamen einfließen. Für die Kategorienbildung des Odds Ratios werden dafür beispielsweise folgende Prädikate konstruiert:

$oddsratio_gleich_eins(< situationsnummer >)$
$kleines_oddsratio(< situationsnummer >)$
$mittleres_oddsratio(< situationsnummer >)$
$gro\ss es_oddsratio(< situationsnummer >)$

Die Repräsentation der anderen schichtunabhängigen Eigenschaften ist völlig analog. Bei den schichtabhängigen Dateneigenschaften wird wieder ein zusätzliches Argument benötigt, z.B. für die Fallanzahlen:

$wenig_faelle(< situationsnummer >, < schichtnummer >)$
$mittelviele_faelle(< situationsnummer >, < schichtnummer >)$
$viele_faelle(< situationsnummer >, < schichtnummer >)$
$sehrviele_faelle(< situationsnummer >, < schichtnummer >)$

In der alternativen Repräsentation wird die Bildung eines Prädikats für jede der schichtunabhängigen Eigenschaften und Bewertungen vorgeschlagen. Für das Odds Ratio sieht die Repräsentation der Kategorien damit folgendermaßen aus:

$oddsratio_kategorie(< situationsnummer >, < kategorie >)$

An der zweiten Stelle des Prädikats „`oddsratio_kategorie`“ stehen dann Konstanten, die die

Zugehörigkeit der Situation zu einer bestimmten Kategorie ausdrücken, z.B. die Konstante „groß“, die andeuten soll, daß das Odds Ratio in der entsprechenden Situation in die Kategorie großes Odds Ratio fällt.
Für die schichtabhängigen Eigenschaften ergeben sich damit dreistellige Prädikate, hier z.B. für die Fallanzahlen:

$$fallanzahl_kategorie(< situationsnummer >, < schichtnummer >, < kategorie >)$$

Für alle anderen Dateneigenschaften und auch für die Bewertungskriterien können diese Repräsentationsweisen analog übernommen werden, indem immer das Argument, in das die konkreten Werte für die Eigenschaften bzw. die Bewertungen geschrieben werden, weggelassen wird (erste Repräsentation) oder durch die Kategorie ersetzt wird (zweite Repräsentation).
Zur Aufteilung in die Kategorien können entsprechende Regeln eingegeben werden, für das Odds Ratio beispielsweise:

```
oddsratio(S,OR) & gt(OR,7) → großes_oddsratio(S)
bzw.
oddsratio(S,OR) & gt(OR,7) → oddsratio_kategorie(S,groß)
```

je nachdem, welche Repräsentation verwendet werden soll. Die Regeln schließen von dem konkreten Wert für das Odds Ratio „OR“ auf die Kategorie, indem der Wert mithilfe der Built–In Prädikate auf seine Größe untersucht wird. Ist der Wert „OR“ hier größer als sieben, so leiten die Regeln das entsprechende Faktum ab.
Da die abgeleiteten Eigenschaften später für das Lernen wichtig sind, ist es ggf. vorteilhaft, die erste Repräsentation zu verwenden. Auf diese Weise kann das Konstantenlernen umgangen werden, das zu Problemen führen kann (siehe [PG 205, 1993]).

4.3.4 Repräsentation der Eignungen und Empfehlungen

Die Umsetzung der Schätzergebnisse wurde bislang auf die Repräsentation der beiden Bewertungskriterien Bias und MSE beschränkt. Im folgenden werden die daraus abzuleitenden Eignungen und Empfehlungen in der Wissensbasis repräsentiert. Dieses erfolgt erst an dieser Stelle, da die Eignung auf den Kategorien für Bias und MSE beruht.
Zunächst wird nun vorgestellt, wie durch die Eingabe einiger Regeln eine Reihenfolge der Schätzer aufgebaut werden kann, diese geeignet zusammengefaßt wird und daraus der beste Schätzer einer Situation abgeleitet wird. Der zweite Teil des Abschnitts beschreibt die Regel-

menge, die die Eignung aus den Kategorieneinteilungen für den Bias und den MSE ableitet. Nimmt man in einem letzten Schritt dann die kategorisierten Eignungen und Empfehlungen zusammen, läßt sich die Empfehlung entsprechend der Eignung in Kategorien einteilen.
Für die Ableitung einer Empfehlung müssen zunächst Rangfolgen der Schätzergebnisse bzgl. der beiden Bewertungskriterien aufgestellt werden. Dafür werden die Werte der Kriterien entsprechend klassifiziert. Diese Klassifikation kann durch Regeln erfolgen, die die autoepistemischen Operatoren `max_of` und `min_of` (siehe 2.2) verwenden, so daß jeweils der größte bzw. der kleinste Wert einer festen Situation „S" festgehalten werden kann, hier beispielsweise für den Bias:

```
situation(S) & max_of(B,bias(S,Z,B),MaxB) → größterbias(B,S)
situation(S) & min_of(B,bias(S,Z,B),MinB) → kleinsterbias(B,S)
```

Damit stehen die kleinsten und größten Werte für den Bias in jeder Situation fest. Die restlichen Werte liegen also im Mittelfeld. Dieses Mittelfeld wird durch die folgende Regel klassifiziert:

```
situation(S) & bias(S,Z,B) & unknown(kleinsterbias(B,S)) &
unknown(größterbias(B,S)) → mittlererbias_1(B,S)
```

Die Regel besagt, daß ein Bias „B" einer Situation „S", der nicht der kleinste und nicht der größte ist, im Mittelfeld liegt. Diese Klassifikation des Mittelfelds wird benötigt, um die weiteren Ränge zu ermitteln. Dazu werden ähnliche Regeln verwendet wie im ersten Schritt:

```
situation(S) & max_of(B,mittlererbias_1(B,S),MaxB) → zweitgrößterbias(B,S)
situation(S) & min_of(B,mittlererbias_1(B,S),MinB) → zweitkleinsterbias(B,S)
```

Der größte Bias des Mittelfelds ist damit der zweitgrößte Bias, der kleinste Bias des Mittelfelds ist der zweitkleinste Bias insgesamt.
Die letzten beiden Schritte werden daraufhin noch einmal wiederholt, um auch den drittgrößten und drittkleinsten Bias zu ermitteln:

```
situation(S) & bias(S,Z,B) & unknown(kleinsterbias(B,S)) &
unknown(größterbias(B,S)) & unknown(zweitkleinsterbias(B,S))
& unknown(zweitgrößterbias(B,S)) → mittlererbias_2(B,S)
```

```
situation(S) & max_of(B,mittlererbias_2(B,S),MaxB) → drittgrößterbias(B,S)
situation(S) & min_of(B,mittlererbias_2(B,S),MinB) → drittkleinsterbias(B,S)
```

Damit ist für den Bias nun eine Reihenfolge der Werte für jede Situation entstanden. Die Regelmenge für den MSE ist völlig analog zu bilden.
Da bislang nur eine Reihenfolge der Werte besteht, das Ziel aber eine Reihenfolge der Schätzer ist, werden nun den Schätzern entsprechend ihren Werten Ränge zugeordnet:

```
bias(S,Z,B) & kleinsterbias(B,S) → rangbias(S,Z,1)
bias(S,Z,B) & zweitkleinsterbias(B,S) → rangbias(S,Z,2)
bias(S,Z,B) & drittkleinsterbias(B,S) → rangbias(S,Z,3)
bias(S,Z,B) & drittgrößterbias(B,S) → rangbias(S,Z,4)
bias(S,Z,B) & zweitgrößterbias(B,S) → rangbias(S,Z,5)
bias(S,Z,B) & größterbias(B,S) → rangbias(S,Z,6)
```

Die erste Regel besagt beispielsweise folgendes:

> Wenn ein Schätzer „Z“ in einer Situation „S“ einen Bias „B“ hat, der in dieser Situation als kleinster Bias klassifiziert wurde, so wird diesem Schätzer in der Situation der Rang 1 zugeordnet.

Ebenfalls analog hierzu wird die Rangfolge bezüglich des MSE gebildet. Damit entstehen die beiden einzelnen Rangfolgen, die im Anschluß zu einer Gesamtrangfolge zusammengefaßt werden. Das Zusammenfassen geschieht mithilfe einer einfachen Regel, die die Summe der beiden einzelnen Ränge durch das Built–In Prädikat `add` berechnet:

```
rangbias(S,Z,RB) & rangmse(S,Z,RM) & add(RB,RM,RS) → gesamtrang(S,Z,RS)
```

Hat ein Schätzer „Z“ in einer Situation „S“ damit beispielsweise den kleinsten Bias (Rang 1) und den größten MSE (Rang 6), so ergibt sich ein Gesamtrang von 7.
Der Schätzer, der den minimalen Gesamtrang in der Situation hat, wird ausgesucht und ist damit der beste Schätzer in dieser Parameterkonstellation. Die Regel, die den Wert für den minimalen Gesamtrang einer Situation sucht, benutzt wieder den autoepistemischen Operator `min_of`:

```
situation(S) & min_of(RS,gesamtrang(S,Z,RS),MRS) → mingesamtrang(MRS,S)
```

Eine weitere Regel muß nun den bzw. die Schätzer mit diesem minimalen Gesamtrang auswählen und als besten Schätzer klassifizieren:

```
gesamtrang(S,Z,RS) & mingesamtrang(RS,S) → besterschätzer(S,Z)
```

Ein im Rahmen der Modellierung bereits erwähntes Problem besteht darin, daß mehrere Schätzer einen minimalen Gesamtrang haben können. Es wird dann der Schätzer ausgewählt, der einen kleineren Wert für den Bias hat. Diese Auswahl kann durch folgende Regel erreicht werden:

```
situation(S) & besterschätzer(S,Z1) & besterschätzer(S,Z2) &
ne(Z1,Z2) & rangbias(S,Z1,RB1) & rangbias(S,Z2,RB2) & le(RB1,RB2)
→ besterschätzerneu(S,Z1)
```

Gibt es mehr als zwei Schätzer, die einen minimalen Gesamtrang besitzen, so gibt es auch für diese Klassifikation mehrere Schätzer. Die Regelmenge muß dann auf ähnliche Weise noch einmal erweitert werden. Aus Gründen der Übersichtlichkeit wird hier auf eine genaue Beschreibung verzichtet. Es wird im folgenden davon ausgegangen, daß nur zwei Schätzer einen minimalen Rang besitzen.
Anschließend wird für jede Situation genau eine Empfehlung eines Schätzers ermittelt, indem, falls es nur einen besten Schätzer gibt, dieser empfohlen wird (erste Regel) oder, falls es in der Situation mehrere solcher Schätzer gibt und einer davon als `besterschätzerneu` klassifiziert worden ist, dieser empfohlen wird (zweite Regel)[4].

```
besterschätzer(S,Z) & unknown(besterschätzerneu(S,Z1)) → empfehlung(S,Z)
besterschätzerneu(S,Z) → empfehlung(S,Z)
```

Damit ist die Umsetzung der Empfehlung abgeschlossen.
Der zweite Teil des Abschnitts diskutiert nun, wie die Eignung eines Schätzers berechnet werden kann. Die Idee besteht darin, den vier Kategorien für Bias und MSE die Zahlen von eins bis vier zuzuordnen (wie es auch schon in Tabelle 4.2 in Abschnitt 4.2.3 durchgeführt worden ist). Diese Zahlenwerte können addiert werden und in eine kategorisierte Gesamteignung umgesetzt werden. Durch folgende Regeln wird der kleinsten Kategorie der Wert 1, der größten der Wert 4 zugeordnet:

[4] Mußte die gerade erwähnte Wiederholung des Vorgangs durchgeführt werden, so sind auch hier weitere Regeln notwendig.

```
sehr_kleiner_bias(S,Z) → bewertung_bias(S,Z,1)
kleiner_bias(S,Z) → bewertung_bias(S,Z,2)
mittlerer_bias(S,Z) → bewertung_bias(S,Z,3)
grosser_bias(S,Z) → bewertung_bias(S,Z,4)
```

Die Bewertung des MSE erfolgt analog. Anschließend werden durch die folgende Regel beide Bewertungen addiert und so zu einer Gesamtbewertung zusammengefaßt:

```
bewertung_bias(S,Z,Bb) & bewertung_mse(S,Z,Bm) & add(Bb,Bm,Bs)
→ bewertung(S,Z,Bs)
```

Im nächsten Schritt müssen die in Tabelle 4.2 aufgestellten Kriterien für die Stufenbildung der Eignung umgesetzt werden. Dazu werden folgende Regeln eingegeben:

```
bewertung(S,Z,B) & eq(B,2) → sehr_gute_eignung(S,Z)
bewertung(S,Z,B) & ge(B,3) & le(B,4) → gute_eignung(S,Z)
bewertung(S,Z,B) & ge(B,5) & le(B,6) → mittlere_eignung(S,Z)
bewertung(S,Z,B) & ge(B,7) → schlechte_eignung(S,Z)
```

Damit hat man die kategorisierte Eignung repräsentiert. In einem letzten Schritt wird mit dieser Eignung auch die Empfehlung bewertet:

```
empfehlung(S,Z) & sehr_gute_eignung(S,Z) → sehr_gute_empfehlung(S,Z)
empfehlung(S,Z) & gute_eignung(S,Z) → gute_empfehlung(S,Z)
empfehlung(S,Z) & mittlere_eignung(S,Z) → mittlere_empfehlung(S,Z)
empfehlung(S,Z) & schlechte_eignung(S,Z) → schlechte_empfehlung(S,Z)
```

Das heißt der beste Schätzer einer Situation wird daraufhin untersucht, wie die Eignung des Schätzers bewertet wurde und entsprechend kategorisiert, da auch der beste Schätzer einer Situation sehr schlecht sein kann.

4.4 Evaluierung des erstellten Modells

Der folgende Abschnitt beschäftigt sich mit der Evaluierung der modellierten Wissensbasis. In diesem Rahmen sollte festgestellt werden, ob die Modellierung des Wissens den Vorstel-

	JK	W_JK	W	MH	JK_ii	BL	insg.
sehr_gute_empfehlung	18	3	1	3	2	1	28
gute_empfehlung	56	8	12	2	17	3	98
mittlere_empfehlung	49	3	13	3	11	9	88
schlechte_empfehlung	9	3	4	6	3	1	26

Tabelle 4.3: Häufigkeiten der Empfehlungen für die Schätzer

	JK	W_JK	W	MH	JK_ii	BL	insg.
sehr_kleiner_bias	88	43	50	53	62	58	364
kleiner_bias	108	106	105	100	104	110	633
mittlerer_bias	23	76	73	73	62	62	369
grosser_bias	21	15	12	14	12	10	74

Tabelle 4.4: Einteilungen der Schätzer für den Bias

lungen des Experten entspricht. Der schwierigste Schritt bei der Erstellung des Modells ist die Einteilung der Dateneigenschaften und Bewertungen in Kategorien. Zur Überprüfung der Kategorienbildung ist die Aufteilung der Situationen der Simulationsstudie in die gebildeten Kategorien zu analysieren. Diese Auswertung wurde nach jeder Änderung der Einteilung erneut durchgeführt, wobei die im weiteren aufgeführten Tabellen mit dem Experten besprochen wurden. Hier wird nun die Kategorieneinteilung gemäß Tabelle 4.1 analysiert, deren Ergebnisse im Rahmen der Evaluation als akzeptabel bewertet wurden.

Zunächst werden die Häufigkeiten für die Prädikate, die einen Schätzer für eine Situation empfehlen, betrachtet. Insgesamt gibt es 240 Empfehlungen, die sich wie in Tabelle 4.3 gezeigt auf die einzelnen Schätzer verteilen. Man kann hier eine deutliche Überlegenheit des Jackknife–Schätzers (JK) feststellen, der jeweils am häufigsten in der sehr guten, guten und mittleren Empfehlung vorkommt.

Diese Überlegenheit zeichnet sich auch schon bei den zugrundeliegenden Prädikaten für die Bewertungskriterien Bias und MSE und den entsprechenden Kategorien ab. Insgesamt gibt es jeweils 1440 Fakten für die Prädikate `bias` und `mse`. Die Verteilung auf die Schätzer ist in den Tabellen 4.4 und 4.5 dargestellt.

Die Analyse der Dateneigenschaften beginnt mit den Einteilungen für die Anzahl der Schichten einer Parameterkonstellation.

Es gibt jeweils 80 Parameterkonstellationen mit 2 (wenigen), 5 (mittelvielen) und 10 (vie-

	JK	W_JK	W	MH	JK_ii	BL	insg.
sehr_kleiner_mse	29	28	28	28	28	28	169
kleiner_mse	74	74	76	74	75	73	446
mittlerer_mse	84	86	85	85	84	86	510
grosser_mse	53	52	51	53	53	53	315

Tabelle 4.5: Einteilungen der Schätzer für den MSE

schichtanzahl	**240**
wenig_schichten	80
mittel_schichten	80
viele_schichten	80
sehr_viele_schichten	0

Tabelle 4.6: Schichtanzahlen

len) Schichten, siehe Tabelle 4.6. Andere Ausprägungen für die Schichtanzahl werden nicht überprüft. Die vorhandenen Anzahlen stehen damit jeweils stellvertretend für eine ganze Kategorie. Da es sich um eine Studie aus dem Large–Strata–Modell handelt, gibt es keine Konstellation mit sehr vielen Schichten. Die Kategorie wurde jedoch mit aufgenommen, um für solche Situationen in einer eventuell durchzuführenden Simulationsstudie mit einem dementsprechenden Design vorbereitet zu sein.

Bei der Einteilung der Größe des Odds Ratios werden alle Kategorien untersucht. Allerdings gibt es in drei Klassen jeweils nur eine Ausprägung, die hier zugrunde liegt. Für die Klasse „`oddsratio=1`“ wird ein Wert von 1, für „`kleines_oddsratio`“ ein Wert von 1.7, für „`mittleres_oddsratio`“ die Werte 3.5 und 5 und für „`großes_oddsratio`“ schließlich ein Wert von 10 überprüft. Für jeden gewählten Wert werden 48 Konstellationen untersucht, so daß es 96 Situationen mit einem mittleren Odds Ratio gibt, siehe Tabelle 4.7.

Für die Fallanzahlen der einzelnen Schichten gibt es 1360 Fakten, aus denen die durchschnittlichen Fallanzahlen, pro Schicht, also 240 Fakten, berechnet werden. Die Einteilung in die Kategorien (siehe Tabelle 4.1) ergibt die in Tabelle 4.8 vorgestellten Ergebnisse. Bei der Einteilung der durchschnittlichen Fallanzahlen zeigt sich, daß zwei Kategorien unbenutzt bleiben, nämlich die für wenige und mittelviele Fälle. Das hängt wiederum mit dem der Simulationsstudie zugrundeliegenden Modell zusammen.

Aus dem Verhältnis von Fällen und Kontrollen pro Schicht (1360 Fakten) wird das durch-

oddsratio	**240**
oddsratio=1	48
kleines_oddsratio	48
mittleres_oddsratio	48
grosses_oddsratio	48

Tabelle 4.7: Odds Ratios

faelle	**1360**
durchschnitt_faelle	**240**
sehr_viele_faelle	150
viele_faelle	90
mittelviele_faelle	0
wenig_faelle	0

Tabelle 4.8: Fallanzahlen

schnittliche Verhältnis berechnet, das sich gemäß Tabelle 4.9 auf die Situationen verteilt. Das häufig vorkommende Verhältnis von 1:3 fällt in die Kategorie „mittel_balanciert“. Alle anderen Verhältnisse gehören zur Kategorie „unbalanciert“. Ein balanciertes Verhältnis tritt nicht auf, da im Design davon ausgegangen wurde, daß in realen Situationen in der Regel wesentlich mehr Kontrollen als Fälle zur Verfügung stehen.

Auch für die durchschnittliche Expositionswahrscheinlichkeit gibt es für eine der drei Kategorien keine Ausprägung in dieser Simulationsstudie und zwar für große Expositionswahrscheinlichkeiten. Diese treten zwar in den einzelnen Schichten auf, der Durchschnitt fällt jedoch nie in diese Kategorie, sondern höchstens in die Kategorie „zentriert“, z.B. wenn die Expositions-

verhaeltnis_kf	**1360**
durchschnitt_verhaeltnis_kf	**240**
unbalanciert_verhaeltnis_kf	40
mittel_balanciert_verhaeltnis_kf	200
balanciert_verhaeltnis_kf	0

Tabelle 4.9: Verhältnis von Fällen und Kontrollen

expo	**1360**
durchschnitt_expo	**240**
kleine_expo	90
zentrierte_expo	150
grosse_expo	0

Tabelle 4.10: Expositionswahrscheinlichkeiten

differenz_expo	**240**
kleine_differenz_expo	150
mittlere_differenz_expo	0
grosse_differenz_expo	90

Tabelle 4.11: Differenzen der Expositionswahrscheinlichkeiten

wahrscheinlichkeit in der ersten von zwei Schichten 0.2 und in der zweiten 0.8 beträgt. Kleine Durchschnittswerte treten dagegen in Situationen mit fünf Schichten auf, wenn z.B. alle fünf Werte zwischen 0.2 und 0.3 liegen. Es ergibt sich die in Tabelle 4.10 gezeigte Aufteilung.
Die Differenz der maximalen und minimalen Expositionswahrscheinlichkeiten einer Schicht ist in 150 Situationen klein, in den weiteren 90 groß (siehe Tabelle 4.11).
In der Beschreibung des Simulationsdesigns (vgl. 3.3.1) wurden drei Konstellationen für die Binomialwahrscheinlichkeiten genannt:

a.) nahe zusammenliegend und kleiner als 0.5,
b.) nahe zusammenliegend und um 0.5 zentriert und
c.) weit auseinanderliegend und um 0.5 zentriert.

Die Konstellationen lassen sich nun durch die folgenden Eigenschaften beschreiben:

a.) kleine_differenz_expo und kleine_expo,
b.) kleine_differenz_expo und zentrierte_expo und
c.) grosse_differenz_expo und zentrierte_expo.

Daraus ist ersichtlich, daß es jeweils 90 Parameterkonstellationen mit den Eigenschaften a.) und c.) und 60 mit der Eigenschaftskombination b.) gibt.

gini_expo	**240**
gini_expo_balanciert	130
gini_expo_mittel_balanciert	80
gini_expo_unbalanciert	30

Tabelle 4.12: Balanciertheit der Expositionswahrscheinlichkeiten

gini_faelle	**240**
gini_faelle_balanciert	150
gini_faelle_mittel_balanciert	90
gini_faelle_unbalanciert	0

gini_verhaeltnis_kf	**240**
gini_verhaeltnis_kf_balanciert	120
gini_verhaeltnis_kf_mittel_balanciert	120
gini_verhaeltnis_kf_unbalanciert	0

Tabelle 4.13: Balanciertheit der Fallanzahlen und des Verhältnisses von Fällen und Kontrollen

Abschließend werden nun noch die Balanciertheitseigenschaften analysiert. Für die Expositionswahrscheinlichkeiten verteilen sich die Situationen wie in Tabelle 4.12 dargestellt auf die drei Kategorien.
Für die beiden anderen Balanciertheitseigenschaften werden keine Situationen in die Kategorie unbalanciert eingeteilt, siehe Tabelle 4.13.

4.5 Lernaufgabe

Der Output der Lernphase soll in einer Charakterisierung der Schätzer bzgl. der kategorisierten Dateneigenschaften bestehen. Dabei werden die schichtabhängigen Dateneigenschaften nicht mehr herangezogen. Weiterhin soll auch die Anzahl der Kontrollen unberücksichtigt bleiben, da diese durch die Eigenschaften Fallanzahl und das Verhältnis von Fällen und Kontrollen schon ausreichend bestimmt ist. Damit verbleiben also neun Dateneigenschaften. Mit der Kategorieneinteilung für diese Eigenschaften ergeben sich 22 unterschiedliche Merkmale der Daten, deren Einfluß auf die Schätzergebnisse festgehalten werden soll.
Die Charakterisierung kann sowohl aus Regeln über die bewerteten Eignungen der Schätzer als auch über die bewerteten Empfehlungen bestehen. Zudem sollte auch über die unbewertete Empfehlung gelernt werden. Damit ergeben sich neun Zielprädikate, über die zu lernen ist.
Charakterisierungen einer Eignung oder Empfehlung können auf unterschiedlich vielen Datenmerkmalen basieren. Durch die Komplexität, die durch die Kombinationsmöglichkeiten der Eigenschaften gegeben ist, sollen die Regeln maximal aus einer Kombination von fünf

Dateneigenschaften aufgebaut werden.

Zur Durchführung der Lernläufe werden geeignete Metaprädikate (siehe 2.2.1) konstruiert. Da die Repräsentation, die den Kategorienamen im Prädikatnamen hat, ausgewählt wurde, gibt es nur einstellige Prädikate für die Dateneigenschaften. Folgende Metaprädikate wurden verwendet:

MP1(S,E1,E): S(Z) & E1(Sit) $\rightarrow$ E(Sit,Z)
MP2(S,E1,E2,E): S(Z) & E1(Sit) & E2(Sit) $\rightarrow$ E(Sit,Z)
MP3(S,E1,E2,E3,E): S(Z) E1(Sit) & E2(Sit) & E3(Sit) $\rightarrow$ E(Sit,Z)
MP4(S,E1,E2,E3,E4,E): S(Z) & E1(Sit) & E2(Sit) & E3(Sit) & E4(Sit) $\rightarrow$ E(Sit,Z)
MP5(S,E1,E2,E3,E4,E5,E): S(Z) & E1(Sit) & E2(Sit) & E3(Sit) & E4(Sit) & E5(Sit) $\rightarrow$ E(Sit,Z)

Mithilfe der Prädikate, durch die das S(Z) instanziiert wird, kann die Variable Z in der Konklusion gebunden werden, siehe Abschnitt 2.2. Mit dem ersten Metaprädikat werden zunächst Dateneigenschaften gesucht, die alleine ausreichen, um einen bestimmten Schätzer zu empfehlen. Anschließend werden Kombinationen von zwei, drei, vier und fünf Eigenschaften gesucht.

Es ist sinnvoll, dieser Modellierung eine Closed World Assumption zugrunde zu legen. Es sollen möglichst wenige bzw. keine neuen Fakten aus den gelernten Regeln gefolgert werden, da alle korrekten Empfehlungen und Eignungen bereits in der Wissensbasis vorliegen. Wird in einer bestimmten Situation ein Schätzer empfohlen, so führt das Ableiten einer weiteren Empfehlung für diese Situation zu einer inkorrekten Wissensbasis. Ebenso ist z.B. das Ableiten einer guten Eignung für einen Schätzer in einer Situation als inkorrekt zu bewerten, wenn für diesen Schätzer hier bereits eine andere Eignung, beispielsweise eine sehr gute Eignung, festgestellt wurde.

Optimal sind also die Regeln mit einem `neg`– bzw. `pred`–Wert von null. Daher wurden zwei Akzeptanzkriterien für die Lernläufe festgelegt, zunächst das sehr strenge Kriterium `pos=total` und ein schwächeres Kriterium mit `pos≥0.9*total`.

Dem Lernvorgang liegt eine Prädikattopologie zugrunde, die zwei Topologieknoten enthält: einen Knoten für die Prädikate, die die Dateneigenschaften beschreiben, und einen Knoten für die Empfehlungs– und Eignungsprädikate. Durch die Metapädikate ist diese Struktur für den Lernvorgang schon gegeben, so daß die Berücksichtigung der Topologie keine weiteren Einschränkungen des Hypothesenraums ergibt.

Desweiteren gibt es zwei Sorten: die Situationen und die Schätzer. Da beide in den Zielprädikaten vorkommen, sind alle Variablen sofort gebunden.

4.6 Lernläufe

Zur Erstellung der Lernläufe wurden die drei Parameter Zielprädikat, Metaprädikat und das Akzeptanzkriterium variiert. Durch die neun Zielprädikate, fünf Metaprädikate und zwei Akzeptanzkriterien ergeben sich 90 durchzuführende Lernläufe. Da sich aber bereits nach den ersten Läufen abzeichnete, daß einige Lernläufe erfolglos bleiben würden, konnte die Anzahl der Lernläufe schließlich auf 31 gesenkt werden. In den Tabellen 4.14 und 4.15 sind die Läufe und ihre Ergebnisse zusammengefaßt.

4.7 Auswahl einer Regelmenge

In diesem Abschnitt wird eine Regelmenge aus den gelernten Regeln zusammengestellt, die später in das wissensbasierte System eingebunden werden soll. Die Regeln unterscheiden sich in ihrer Güte, so daß hier nur die „besten" Regeln ausgewählt werden sollen. Für diese Auswahl, die im folgenden beschrieben wird, spielt die Aussagekraft der Konklusionsprädikate sowie die Redundanz innerhalb der Regelmenge eine Rolle.
Da schon für die Empfehlungsprädikate eine ganze Reihe von Regeln gelernt werden konnten, und diese eine höhere Aussagekraft besitzen als die Regeln für die Eignungsprädikate, werden ausschließlich diese für die „endgültige" Regelmenge in Betracht gezogen.
Da für das Metapädikat MP1 keine Regeln gelernt werden konnten, enthält die Auswahlmenge keine Regeln, die von einer Dateneigenschaft auf die Empfehlung eines Schätzers schließen. Im folgenden werden nun die mit den Metaprädikaten MP2 bis MP4 gelernten Regeln betrachtet.
Das größte Problem bei der Regelauswahl ist hier die Redundanz in der Regelmenge (siehe auch 4.9.3). Häufig decken verschiedene Regeln dieselben Situationen ab. Diese Redundanzen können mithilfe des Tools RRT entdeckt werden. Zur Verdeutlichung betrachte man folgendes Beispiel:

Man hat die Regeln:

1. oddsratio=1(S) & viele_faelle(S) & wenig_schichten(S)
   → mittlere_empfehlung(jk_ii,S)
2. oddsratio=1(S) & viele_faelle(S) & gini_faelle_balanciert(S)
   → mittlere_empfehlung(jk_ii,S)

Die beiden Regeln unterscheiden sich nur in der dritten Prämisse. Durch jede Regel werden die Parameterkonstellationen eins (`sit_1`) bis sechs (`sit_6`) abgedeckt. Grund hierfür ist ein Zusammenhang zwischen den Dateneigenschaften `wenig_schichten` und `gini_faelle_balanciert` in der Simulationsstudie. Es

Zielprädikat	**MP**	**AK**	**Ergebnis / Bemerkungen**
alle	MP1	alle	keine gelernten Regeln
`sehrgute_empfehlung`	MP2	pos>0.9*total pos=total	keine gelernten Regeln
`gute_empfehlung`	MP2	pos=total	eine gelernte Regel Lernzeit: 3343 Sekunden Regel für Jackknife Schätzer JK
`mittlere_empfehlung`	MP2	pos>0.9*total pos=total	keine gelernten Regeln
`schlechte_empfehlung`	MP2	pos>0.9*total pos=total	keine gelernten Regeln
`empfehlung`	MP2	pos=total	acht gelernte Regeln Lernzeit: 7137 Sekunden Regeln für Jackknife Schätzer JK
`empfehlung`	MP2	pos>0.9*total	10 Regeln Lernzeit: 8452 Sekunden Regeln für Jackknife Schätzer JK
`sehrgute_eignung`	MP2	pos=total	20 gelernte Regeln Lernzeit: 6492 Sekunden keine Regel für Breslow–Liang Schätzer Eigenschaft „oddsratio=1“ kommt in jeder Regel vor
`gute_eignung`	MP2	pos>0.9*total	fünf gelernte Regeln Lernzeit: 6754 Sekunden Regeln für alle Schätzer außer Woolf Jackknife
`gute_eignung`	MP2	pos=total	keine gelernten Regeln
`mittlere_eignung`	MP2	pos>0.9*total	zwei gelernte Regeln Lernzeit: 6129 Sekunden Regeln für Jackknife II Schätzer JK_ii und Breslow–Liang
`mittlere_eignung`	MP2	pos=total	keine gelernten Regeln
`schlechte_eignung`	MP2	pos>0.9*total pos=total	keine gelernten Regeln

Tabelle 4.14: Lernläufe

Zielprädikat	**MP**	**AK**	**Ergebnis / Bemerkungen**
`sehrgute_empfehlung`	MP3	pos>0.9*total pos=total	keine gelernten Regeln
`gute_empfehlung`	MP3	pos=total	neun gelernte Regeln Lernzeit: 30044 Sekunden Regeln für Jackknife Schätzer JK
`gute_empfehlung`	MP3	pos>0.9*total	18 gelernte Regeln Lernzeit: 31223 Sekunden Regeln für Jackknife Schätzer JK
`mittlere_empfehlung`	MP3	pos=total	24 gelernte Regeln Lernzeit: 32678 Sekunden Regeln für Jackknife Schätzer JK
`schlechte_empfehlung`	MP3	pos=total	eine gelernte Regel Lernzeit: 29452 Sekunden Regel für Mantel–Haenszel Schätzer
`sehrgute_eignung`	MP3	pos=total	ca. 750 gelernte Regeln nicht vollständig ausgewertet
alle Eignungsprädikate	MP3	alle	nicht durchgeführt[a]
`sehrgute_empfehlung`	MP4	pos=total	keine gelernten Regeln
`gute_empfehlung`	MP4	pos=total	ca. 80 Regeln Regeln für Jackknife und Woolf Jackknife Schätzer
`mittlere_empfehlung`	MP4	pos=total	ca. 100 Regeln Lernlauf wurde abgebrochen[b]
`schlechte_empfehlung`	MP4	pos=total	12 Regeln Regeln für Mantel Haenszel Schätzer
alle	MP5	alle	konnten nicht durchgeführt werden[c]

Tabelle 4.15: Lernläufe

[a] Hier wurde bereits entschieden, daß diese Prädikate nicht weiter betrachtet werden sollen, siehe 4.7.

[b] Der Lauf wurde nach ca. einwöchiger Laufzeit abgebrochen.

[c] Die Lernläufe konnten aufgrund von Effizienzproblemen (siehe oben) nicht durchgeführt werden.

Odds Ratios	1
Schichtanzahl	2
Fallanzahl	3
Exposition	4
Differenz der Expos.	5
Verhältnis von Kontrollen und Fällen	6
Balanciertheit der Fallanzahlen	7
Balanciertheit der Expositionen	8
Balanciertheit der Verhältisse	9

Tabelle 4.16: Bewertung / Reihenfolge der Dateneigenschaften

wurde im Simulationsdesign festgelegt, daß bei einer kleinen Anzahl an Schichten stets gleichmäßig über die Schichten verteilte Fallanzahlen vorliegen. Umgekehrt gilt auch, daß gleichmäßig verteilte Fallanzahlen nur in Situationen mit wenig Schichten vorkommen. Daraus folgt, daß beide Prämissen die gleiche Bedeutung haben.

Eine ganze Reihe solcher Zusammenhänge haben dazu geführt, daß sehr viele redundante Regeln gelernt wurden. Hier wird nun eine Auswahl vorgestellt, die jeweils nur eine Regel aus einer solchen redundanten Menge in die endgültige Regelmenge aufnimmt. Das Kriterium für die Auswahl ist eine Bewertung der Dateneigenschaften gemäß Tabelle 4.16, die folgendermaßen eingesetzt werden kann: gibt es redundante Regeln, die die gleichen Situationen abdecken, so werden die Bewertungen der Prämissen addiert, und die Regel mit der niedrigsten Gesamtbewertung wird ausgewählt. Im obigen Beispiel hat die erste Regel eine Bewertung von 1+3+2=6 und für die zweite Regel ergibt sich 1+3+7=11. Damit wird die erste Regel in die Auswahlmenge aufgenommen.
Für das Metaprädikat MP2 werden auf diese Weise fünf Regeln ausgewählt:

```
unbalanciert_verh_kf(S) & viele_schichten(S)
→ gute_empfehlung(S,jk)

kleines_oddsratio(S) & unbalanciert_verh_kf(S)
→ empfehlung(S,jk)

kleine_expo(S) & wenig_schichten(S)
```

→ empfehlung(S,jk)

gini_verh_kf_balanciert(S) & oddsratio_gleich_eins(S)
→ empfehlung(S,jk)

gini_expo_unbalanciert & kleines_oddsratio(S)
→ empfehlung(S,jk)

Für das Metaprädikat MP3 ergeben sich die folgenden sechs Regeln:

oddsratio_gleich_eins(S) & viele_faelle(S) & wenig_schichten
→ mittlere_empfehlung(S,jk)

kleines_oddsratio(S) & viele_faelle(S) & wenig_schichten
→ mittlere_empfehlung(S,jk)

mittel_schichten(S) & oddsratio_gleich_eins(S) & viele_faelle(S)
→ gute_emfehlung(S,jk)

viele_schichten(S) & oddsratio_gleich_eins(S) & viele_faelle(S)
→ gute_emfehlung(S,jk)

wenig_schichten(S) & oddsratio_gleich_eins(S) & sehrviele_faelle(S)
→ gute_emfehlung(S,jk)

gini_faelle_mittelbalanciert(S) & grosse_differenz_expo(S) &
grosses_oddsratio(S)
→ schlechte_empfehlung(S,mh)

Schließlich gibt es noch sieben Regeln, die von einer Kombination von vier Dateneigenschaften auf die Empfehlung eines Schätzers schließen:

viele_schichten(S) & kleines_oddsratio(S) & viele_faelle(S)
& kleine_expo(S) → gute_empfehlung(S,jk)

mittelviele_schichten(S) & kleines_oddsratio(S) & kleine_expo(S)

```
& viele_faelle(S) → mittlere_empfehlung(S,jk)

mittelviele_schichten(S) & mittleres_oddsratio(S) & kleine_expo(S)
& viele_faelle(S) → mittlere_empfehlung(S,jk)

mittelviele_schichten(S) & kleines_oddsratio(S) & kleine_expo(S)
& sehrviele_faelle(S) → gute_empfehlung(S,jk)

viele_schichten(S) & grosses_oddsratio(S) & viele_faelle(S)
& zentrierte_expo(S) → schlechte_empfehlung(S,mh)

viele_schichten(S) & kleines_oddsratio(S) & viele_faelle(S)
grosse_differenz_expo(S) → gute_empfehlung(S,w_jk)

viele_schichten(S) & grosses_oddsratio(S) & viele_faelle(S)
kleine_expo(S) → empfehlung(S,jk_ii)
```

Da die Lernläufe mit dem Metaprädikat MP5 nicht erfolgreich abgeschlossen werden konnten, wurde hier eine Regelüberprüfung „per Hand" durchgeführt. Aus den Traces der Lernläufe mit dem Metaprädikat MP4 konnten einige Vermutungen über mögliche Regeln aufgestellt werden. Diese wurden dann in das System eingegeben und untersucht, wieviele Fakten neu abgeleitet werden. Diese Anzahl wurde mit der Anzahl der Situationen verglichen, die insgesamt abgedeckt werden. Von den überprüften Regeln wurden folgende ausgewählt:

```
wenig_schichten(s) & grosses_oddsratio(S) & sehrviele_faelle(S)
& zentrierte_expo(S) & kleine_differenz_expo(S) → gute_empfehlung(S,jk)

viele_schichten(S) & grosses_oddsratio(s) & sehrviele_faelle(S)
& zentrierte_expo(S) & kleine_differenz_expo(S) → mittlere_empfehlung(S,jk)

wenig_schichten(s) & grosses_oddsratio(S) & viele_faelle(S)
& unbalanciert_verh_kf(S) & grosse_differenz_expo(S)
→ schlechte_empfehlung(S,w_jk)

mittelviele_schichten(s) & zentrierte_expo(S) & kleine_differenz_expo(S)
& grosses_oddsratio(S) & viele_faelle(S) → schlechte_empfehlung(S,jk_ii)
```

```
wenig_schichten(S) & grosses_oddsratio(S) & viele_faelle(S)
mittel_balanciert_verh_kf(S) & unknown(kleine_expo(S))
→ schlechte_empfehlung(S,woolf)
```

Für diese Regeln gibt es keine negativen bzw. vorhergesagten Empfehlungsfakten.
Im Rahmen dieser Überprüfung ergab sich auch die Regel

```
wenig_schichten(S) & grosses_oddsratio(S) & viele_faelle(S)
& unbalanciert_verh_kf(S) & kleine_differenz_expo(S) & zentrierte_expo(S)
→ schlechte_empfehlung(S,bl)
```

die durch das Hinzufügen einer Eigenschaft entstanden ist. Hierfür sind ebenfalls keine Gegenbeispiele vorhanden. Daher wird auch diese in die Auswahlmenge übernommen.
In der durch die Überprüfung von Regeln entstandenen Regelmenge ergeben sich natürlich keine Redundanzen, da mit den „besten“ Eigenschaften begonnen und bereits beschriebene Situationen nicht weiter betrachtet wurden.
Die ausgewählte Regelmenge besteht damit aus 24 Regeln, 16 Regeln für den Jackknife Schätzer JK, jeweils zwei Regeln für Jackknife–Schätzer JK_ii, den Mantel–Haenszel–Schätzer und den gejackkniften Woolf–Schätzer sowie jeweils eine Regel den Breslow–Liang und den Woolf–Schätzer.

4.8 Bewertungskriterien

In diesem Abschnitt werden die Bewertungskriterien vorgestellt, die im Rahmen der Analyse der gelernten Regeln Verwendung finden.
Das erste Kriterium ist die **Vollständigkeit der Regelmenge**. Üblicherweise wird die Vollständigkeit definiert als

$$\frac{|tat_ext(c) \cap ext(c)|}{|tat_ext(c)|},$$

wobei $ext(c)$ die Menge aller Beispiele ist, die durch das Lernergebnis als c klassifiziert werden und $tat_ext(c)$ die Menge der tatsächlichen Instanzen des Begriffs c bezeichnet.
Unter $tat_ext(c)$ kann hier nur die Menge der tatsächlichen Instanzen des Begriffs c bzgl. der zugrundeliegenden Simulationsstudie verstanden werden. Vollständigkeit bezeichnet daher hier das Verhältnis der durch die Regeln abgedeckten Beispiele und den insgesamt in der Wissensbasis zum Lernen zur Verfügung gestellten Beispiele.

Die **Korrektheit der Regelmenge** ist definiert durch

$$1 - \frac{|not(tat_ext(c)) \cap ext(c)|}{|ext(c)|},$$

d.h. wenn keine Fakten im Widerspruch zur tatsächlichen Extension abgeleitet werden, ist die Regelmenge korrekt.

Da in diesem Lernszenario eine Closed World Assumption zugrunde gelegt worden ist, stehen alle von der Regelmenge neu abgeleiteten Fakten im Widerspruch zur tatsächlichen Extension.

Ein weiteres wichtiges Kriterium ist die **Redundanz**.

> Eine Regel R ist redundant bzgl. R', wenn die Menge der Extensionen von R eine Teilmenge der Menge der Extensionen von R' ist.

Außer dieser extensional definierten Redundanz gibt es auch die intensionale Redundanz oder logische Redundanz, die sich folgendermaßen bestimmen läßt:

> Eine Regel R ist intensional redundant bzgl. R', wenn die Prämissen von R logisch aus den Prämissen von R' gefolgert werden können.

Einfacher ist die Festlegung des Kriteriums **Länge der Regeln** bzw. die **Größe der Regelmenge**, bei dem jeweils die Anzahl der Prämissen gezählt wird.

Desweiteren wird die Anzahl der von der Regelmenge insgesamt **abgedeckten Instanzen** zur Bewertung herangezogen.

Hiermit verbunden ist auch der **Komprimierungsgrad** der Regeln. Die Berechnung dieses Kriteriums, wie sie in RRT durchgeführt wird (siehe [Morik et al, 1993]), scheint in der vorliegenden Situation weniger sinnvoll. Daher wird der Komprimierungsgrad hier definiert als

$$\frac{|Prämissen|}{|abgedeckte\ Instanzen|}.$$

Ein weiteres gebräuchliches Kriterium ist die Lernzeit, die nicht zur Bewertung der Regeln verwendet wird, da es in dieser Arbeit weniger um eine Bewertung des Lernverfahrens geht als um die gelernten Regeln selber. Wichtiger erscheint in diesem Zusammenhang eher die Zeit, die insgesamt, also inklusive Modellierung, Repräsentation etc., benötigt wurde, um die Regeln zu erhalten. Hier ist eine objektive Zeitmessung jedoch nahezu unmöglich.

4.9 Analyse der gelernten Regeln

In den folgenden Abschnitten werden die Regeln ausgewertet, die in dem oben beschriebenen Lernszenario entdeckt werden konnten. Dabei werden die in 4.8 aufgelisteten Bewertungskriterien zugrunde gelegt.

4.9.1 Vollständigkeit

Die aus den in 4.7 beschriebenen 24 Regeln bestehende Regelmenge deckt insgesamt 40% der Beispiele ab. Damit gibt es also für 97 der 240 Parameterkonstellationen eine Beschreibung durch eine Regel, die einen Schätzer für diese Situation empfiehlt.

Die Abdeckung von 40% könnte auf einfache Weise erhöht werden. Dazu müßten die Kategorieneinteilungen geändert werden, so daß weniger Werte für die Parameter in einer Kategorie zusammengefaßt werden. Wenn mehrere Ausprägungen einer Dateneigenschaft in einer Kategorie existieren, können Situationen, die sich nur durch diese Eigenschaft unterscheiden, nicht mehr getrennt werden. Gibt es beispielsweise zwei Situationen, die sich lediglich durch die Anzahl der Fälle insofern unterscheiden, als die eine Situation beispielsweise 20, die andere durchschnittlich 30 Fälle pro Schicht hat und fallen 20 und 30 in eine Kategorie, so sind die Beschreibungen der Situationen gleich. Kommt es also vor, daß in solchen Situationen unterschiedliche Schätzer am besten geeignet waren, kann hierfür keine Regel gelernt werden.

Durch eine feinere Kategorieneinteilung ist eine Vollständigkeit von 100% erreichbar, spätestens dann, wenn jede Kategorie einer in der Simulationsstudie untersuchten Ausprägung für die Dateneigenschaft entspricht. Eine solche Einteilung ist natürlich nicht sinnvoll.

Die Kategorieneinteilung wurde im Anschluß an diese Auswertung noch einmal überprüft, insbesondere die Einteilungen für das mittlere Odds Ratio und für große Differenzen der Expositionswahrscheinlichkeiten. Bezüglich der Expositionswahrscheinlichkeiten könnten beispielsweise durch eine Einteilung in große und sehr große Differenzen, die die Werte 0.9 und 0.6 trennt, eine Reihe von zusätzlichen Regeln gelernt werden. Dieser Vorschlag wurde jedoch von den Experten abgelehnt, da die bestehende Einteilung als adäquater betrachtet wurde.

Insgesamt läßt sich hierzu sagen, daß der Sachbereich auf die Veränderung der Kategorieneinteilung sehr sensibel reagiert. Das zeigt sich darin, daß auch relativ geringe Änderungen der Dateneigenschaften das Verhalten der Schätzer stark beeinflussen können. Wenn aber Kategorien gebildet werden, die die leichten Variationen der Dateneigenschaften in unterschiedliche Kategorien teilen, so steht in dieser Simulationsstudie am Ende jeweils nur noch eine untersuchte Ausprägung stellvertretend für eine ganze Kategorie. Daraus ergibt sich die Forderung nach umfassenderen Simulationsstudien, so daß auch für solche „kleinen“ Kategorien noch eine Reihe unterschiedlicher Ausprägungen untersucht werden, z.B. weitere Ausprägungen für das Odds Ratio zwischen 1 und 1.7. Für Simulationsstudien, die nicht mit diesem Ansatz ausgewertet werden und lediglich einen groben Vergleich von Schätzern liefern sollen, sind die gewählten Ausprägungen ausreichend. Um aber Empfehlungen bzw. Charakterisierungen zu lernen, wären wesentlich umfangreichere Simulationen erforderlich.

Weitere Überlegungen betreffen die Vollständigkeit der Regelmenge in der Anwendung auf

„reale“ Fall–Kontroll–Studien. Es stellt sich dabei die Frage, wie oft der Fall eintritt, daß keine Regel angewendet werden kann, weil die Dateneigenschaften durch keine Regel abgedeckt werden können, d.h. es ist die Frage, wieviele mögliche Datensituationen[5] nicht abgedeckt werden können. Es ergibt sich hier theoretisch eine sehr große Anzahl. Dabei muß aber auch berücksichtigt werden, wie oft diese Konstellationen der Daten in der „Praxis“ tatsächlich vorkommen.

Untersucht man beispielsweise die Datensituationen mit einer kleinen Anzahl von Fällen, so stellt man fest, daß es keine Regel für diese Eigenschaft gibt. Solche Situationen kommen in der praktischen Anwendung aber eher seltener vor als Situationen mit vielen Fällen. Zudem werden Studien mit kleinen Fallanzahlen nicht unbedingt mittels einer Kontingenztafelanalyse ausgewertet.

Da die Simulationstudien gerade für die Praxis relevante Situationen untersuchen sollten, kann hier also mit relativ guten Werten gerechnet werden, auch wenn es theoretisch sehr viele nicht abgedeckte Situationen gibt. Weiter ist zu beachten, daß Situationen mit nicht erfaßten Eigenschaften trotzdem durch eine Regel abgedeckt werden können. Für eine Situation mit wenig Fällen sind beispielsweise nur die Regeln, die sich explizit auf die Fallanzahl beziehen nicht anwendbar. Andere Regeln, die keine bestimmte Fallanzahl voraussetzen, können angewendet werden.

Aus diesen Überlegungen ergibt sich auch ein weiteres Kriterium zur Bewertung der Regeln, nämlich der Stellenwert der Regeln in der praktischen Anwendung. Zur Beurteilung sind hier die ersten mit dem System durchgeführten Analysen heranzuziehen.

4.9.2 Korrektheit

Da fast alle der ausgewählten Regeln mit dem Akzeptanzkriterium `pos=total` gelernt wurden, gibt es lediglich sieben durch die Regeln abgeleiteten Empfehlungen, die nicht schon in der Wissensbasis vorgelegen haben. Es wird also eine Korrektheit von ca. 93% erreicht.

Aber auch hier ist es wieder interessant, die Ergebnisse in der praktischen Anwendung zu prüfen.

Werden für eine Kategorie nur sehr wenige Werte in der Simulationsstudie untersucht, so ist die Korrektheit der Regel für die gesamte Kategorie fraglich. Das folgende Beispiel soll dies verdeutlichen:

> Für die Eigenschaft „kleines Odds Ratio“ liegt die untere Grenze dieser Kategorie bei 1 und schließt Werte bis zu 2 ein. In der Simlationsstudie wird aber lediglich der Wert 1.7 überprüft. Da keine Situationen mit einem Odds Ratio von 1.8 oder 2

[5] beschrieben durch die kategorisierten Dateneigenschaften, nicht die konkreten Werte

untersucht werden, darf eine Korrektheit von 93% in der praktischen Anwendung der Regeln sicher nicht angenommen werden.

Eine übliche Vorgehensweise ist, die Korrektheit durch eine Auswertung der ersten Anwendungen des Systems zu untersuchen. Dieses Vorgehen ist hier nicht möglich, da die Korrektheit in der realen Anwendung nicht beurteilt werden kann. Die zwei Möglichkeiten, die sich anbieten, sind die Einteilung der vorhandenen Simulationsstudie in ein Lern– und ein Testset (siehe 2.2) oder die Durchführung weiterer Simulationsstudien, da hier die Korrektheit überprüft werden kann. Die Einteilung in ein Lern– und Testset hätte vermutlich keine wesentlichen Veränderungen der Bewertungen erbracht. Der Mehraufwand für die Durchführung der Lernläufe ist aber erheblich, wenn man z.B. eine 240–fache Cross Validation zugrunde legt, so daß jeweils ein Beispiel als Testset fungiert. Eine Cross Validation, die ein Lern– und Beispielset mit beispielsweise jeweils 120 Situationen vorsieht und daher weniger Aufwand bedeutet, kann hier nicht durchgeführt werden, da es nicht in erster Linie um eine Bewertung der Regeln geht, sondern die Regeln selbst von Bedeutung sind und höchstens geringe Qualitätsverluste in Kauf genommen werden können. Die zweite vorgeschlagene Möglichkeit, also die Validierung durch weitere Simulationsstudien, scheint daher sinnvoller.

4.9.3 Redundanz

Die Extensionen der 24 ausgewählten Regeln sind paarweise disjunkt, d.h. es gibt keine extensionale Redundanz in der Regelmenge. Wie in Abschnitt 4.7 beschrieben wurde, ist aus den Mengen redundanter Regeln jeweils nur eine ausgewählt worden.
Die Redundanz bezieht sich auch hier wieder nur auf die Beispielmenge, nicht auf eine mögliche Redundanz bei der Anwendung der Regeln.
Das Kriterium für die Auswahl der Regeln ist eine Bewertung der Dateneigenschaften. In Bezug auf die reale Anwendung der Regeln sind diese Entscheidungen allerdings mit Unsicherheit behaftet, da sich natürlich auch die anderen Eigenschaften, die in der Simulationsstudie nicht variiert wurden und damit redundant waren, auf das Verhalten der Schätzer auswirken können. Die Folge ist, daß es bei der Anwendung dieser Regeln zu Problemen kommen könnte. Daher ist zu überlegen, ob es wirklich sinnvoll ist, die redundanten Regeln nicht weiter zu berücksichtigen, da ihre Prämissen zusätzliche Informationen bieten können (siehe 4.9.4).

4.9.4 Länge der Regeln

Die durchschnittliche Länge der Regeln beträgt 3.6 Prämissen pro Regel. Die gesamte Regelmenge enthält 87 Prämissen. Die Anzahl der Variablen und auch der Konstanten pro Regel beträgt jeweils eins.

Allgemein gilt natürlich, daß generellere Regeln vorteilhaft sind, da sie mehr Instanzen abdecken können. Aber wegen der in 4.9.3 angesprochenen Bedenken ist es ggf. vorzuziehen, die unterschiedlichen Prämissen der redundanten Regeln alle zu einer Regel zusammenzufassen. Man betrachte dazu das folgende Beispiel:

> Die beiden Regeln aus Abschnitt 4.7, also
>
> 1. oddsratio=1(S) & viele_faelle(S) & wenig_schichten(S)
>   → mittlere_empfehlung(jk_ii,S)
> 2. oddsratio=1(S) & viele_faelle(S) & gini_faelle_balanciert(S)
>   → mittlere_empfehlung(jk_ii,S)
>
> kann man zu der Regel
>
> oddsratio=1(S) & viele_faelle(S) & wenig_schichten(S) & gini_faelle_balanciert(S)
> → mittlere_empfehlung(jk_ii,S)
>
> zusammenfassen. Diese Regel enthält vier Prämissen und ist spezieller als die beiden Regeln einzeln.

Der Vorteil dieser längeren Regeln ist, daß ihre Korrektheit in der realen Anwendung vermutlich besser ist, da sie berücksichtigen, daß die Zusammenhänge zwischen bestimmten Dateneigenschaften, die in der Simulationsstudie gelten, in der Anwendung der Regeln auf „echte" Daten nicht mehr gegeben sind. Die Regelmenge wird aber zunächst nicht geändert.

4.9.5 Abgedeckte Instanzen

Im Durchschnitt werden pro Regel vier Instanzen abgedeckt, d.h. eine durchschnittliche Regel beschreibt vier Parameterkonstellationen aus der Simulationsstudie. Das Akzeptanzkriterium `pos=total` erlaubt, Regeln bereits aus einem Beispiel zu lernen. Das Kriterium wird so niedrig angesetzt, da gleiche Parameterkonstellationen in der Simulationsstudie nicht vorkommen. Es muß aber bedacht werden, daß für jede Konstellation 1000 Läufe durchgeführt worden sind und das Ergebnis der Mittelwert aus diesen 1000 Simulationsläufen ist. Damit liegen den Beispielen, aus denen gelernt wird, jeweils 1000 Simulationsläufe zugrunde, d.h. die Informationen in den einzelnen Beispielen weisen selbst einen relativ hohen Grad an Sicherheit auf.

4.9.6 Komprimierungsgrad

Die Regelmenge umfaßt 87 Prämissen und deckt damit 97 Instanzen ab, was nach Abschnitt 4.8 einem Komprimierungsgrad von 0.9 entspricht. Von einer echten Komprimierung der Beispiele kann also nicht gesprochen werden. Gerade das war aber auch nicht das Ziel, sondern es sollten mit den Beispielen konsistente Regeln zur Vorhersage weiterer Situationen gelernt werden.

4.10 Einbinden der Regeln in das WBS

Dieser Abschnitt beschäftigt sich damit, wie die gelernten Regeln zur Charakterisierung der Schätzer des gemeinsamen Odds Ratios in das WBS eingebunden werden können.
Das Ziel besteht darin, eine Beratungskomponente zu entwickeln, die den Benutzer des WBS bei der Auswahl eines geeigneten Odds Ratio Schätzers für seine Fall–Kontroll–Studiendaten unterstützt. Zentrales Element dieser Beratungskomponente ist die gelernte Regelmenge. Zur Anwendung der Regeln sind zunächst die Daten des Benutzers auf die festgelegten Dateneigenschaften zu überprüfen. Kann daraufhin eine „passende“ Regel gefunden werden, so wird dem Benutzer ein geeigneter Schätzer vorgeschlagen. „Passend“ bedeutet hier, daß die Prämissen der Regel mit den festgestellten Dateneigenschaften übereinstimmen.
Für die Entwicklung dieser Beratungskomponente müssen die Regeln in das wissensbasierte System einkodiert werden. Dabei sind folgende Punkte wichtig:

- die Auswahl einer geeigneten Regelmenge,
- das Festlegen einer effizienten Abfragereihenfolge für die Dateneigenschaften und
- die Bewertung der Regeln innerhalb der ausgewählten Menge.

Der erste Punkt wurde bereits in Abschnitt 4.7 beschrieben. Grundlage für die weiteren Betrachtungen bildet also die dort ausgewählte Regelmenge bestehend aus 24 Regeln.
Das Ziel der Festlegung einer Abfragereihenfolge ist es, die Anzahl der Abfragen zu minimieren. Dazu muß beachtet werden, daß grundsätzlich alle Prämissen einer Regel abgefragt werden müssen. Da viele Regeln aber teilweise gleiche Prämissen haben, ist es sinnvoll, die am häufigisten vorkommenden Dateneigenschaften zuerst abzufragen. Dazu müssen die Prämissen der Regelmenge untersucht werden. In Tabelle 4.17 sind die Häufigkeiten für die Dateneigenschaften dargestellt.
Demnach wird die Größe des Odds Ratios als erstes abgefragt. Für jede Kategorie müssen dann die Häufigkeiten der weiteren Dateneigenschaften überprüft werden. Für großes Odds Ratio ergibt sich z.B., daß in acht der neun Regeln mit der Prämisse `großes_oddsratio`

Odds Ratio	22
Schichtanzahl	20
Fallanzahl	18
Exposition	12
Differenz Expos.	7
Verhältnis K/F	5
Balance Expos.	1
Balance Verhältnis	1
Balance Fälle	1

Tabelle 4.17: Häufigkeiten der Dateneigenschaften

die Fallanzahl eine Rolle spielt. Da keine andere Eigenschaft häufiger vorkommt, soll die Fallanzahl als zweite Eigenschaft abgefragt werden.

Eine der Regeln mit der Prämisse `großes_oddsratio` berücksichtigt die Fallanzahl nicht. Auf der gleichen Ebene wie die Abfrage der Kategorien für die Fallanzahl können also die Eigenschaften, die in dieser Regel vorkommen, überprüft werden.

Auf diese Weise wird die gesamte Regelmenge analysiert. Man erhält damit den ausschnittsweise in der Abbildung 4.3 vorgestellten Entscheidungsbaum, der die Abfragereihenfolge der Dateneigenschaften festlegt.

Im Rahmen der Anwendung der Regeln kann es vorkommen, daß mehrere Regeln auf eine Datensituation angewendet werden können, d.h. die Situation erfüllt die Prämissen mehrerer Regeln. Zwar ist die extensionale Redundanz bezüglich des Beispielsets ausgeschlossen (siehe 4.9.3), aber da die Regeln nur einen Teil der Dateneigenschaften abfragen, ist es nicht ausgeschlossen, daß auf diese Weise für eine Situation unterschiedliche, d.h. sich widersprechende, Empfehlungen von Schätzern zustande kommen können.

Stellt sich in der Praxis heraus, daß es viele dieser Situationen gibt, ist das ein Hinweis darauf, daß die Regelmenge nicht optimal ist. Der Grund dafür, daß solche Regeln gelernt werden, liegt darin, daß in der Simulationsstudie nicht alle Kombinationen von Eigenschaften untersucht werden, die in der Anwendung auftreten.

Der in Abbildung 4.3 dargestellte Entscheidungsbaum legt die Reihenfolge der Knoten auf gleicher Ebene nicht fest. Daher kann zusätzlich die Bewertung der Regeln aus Abschnitt 4.9 berücksichtigt werden, so daß in den nicht eindeutigen Situationen die als besser bewerteten Regeln bevorzugt werden. Als Gütekriteruim soll hier insbesondere die Anzahl der positiven Beispiele herangezogen werden, die von den Regeln eines unter einem solchen Knoten liegen-

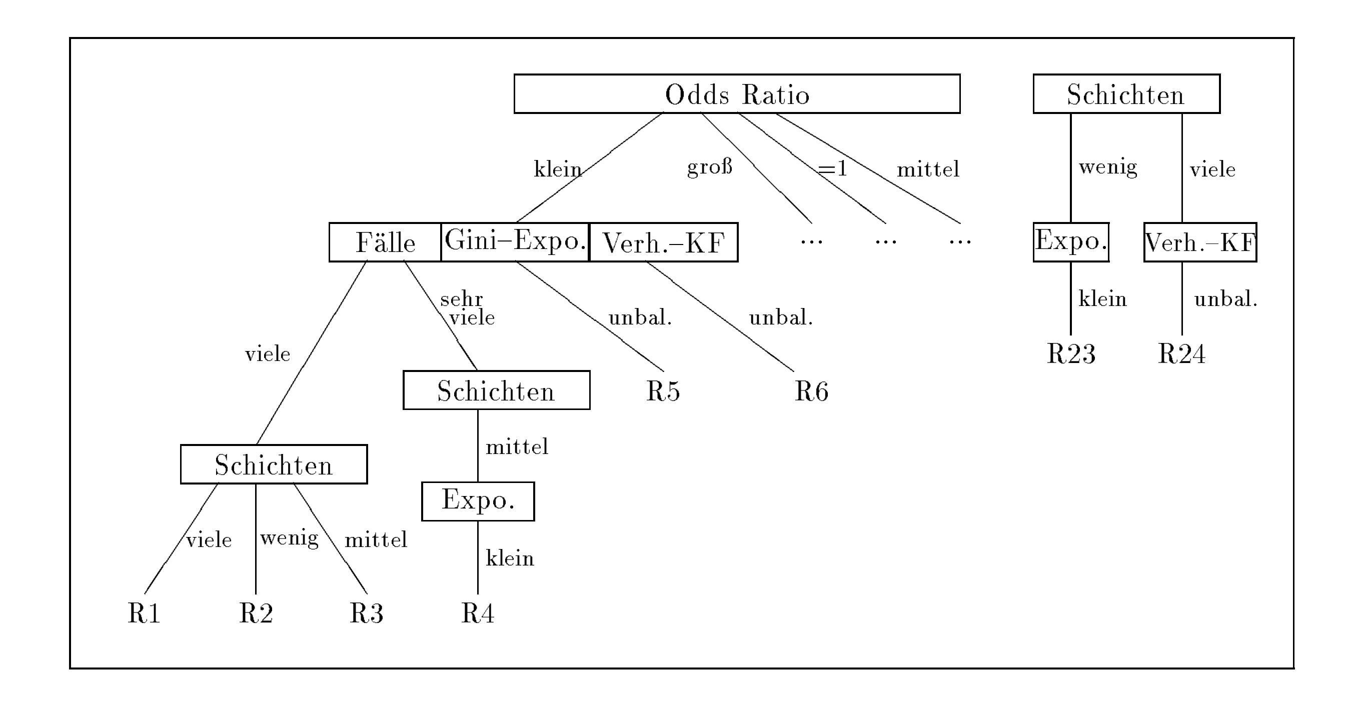


Abbildung 4.3: Entscheidungsbaum

den Astes des Entscheidungsbaums abgedeckt werden. Der Entscheidungsbaum soll deshalb hier von „links nach rechts“ durchsucht werden. Falls eine anwendbare Regel gefunden wird, wird die Suche abgebrochen.

Kapitel 5

Das Symbol Level Modell des wissensbasierten Systems CORA

Dieses Kapitel beschreibt die Operationalisierung der Knowledge Level Modelle, also die Umsetzung des konzeptuellen Modells in das fertige System. Das Ergebnis ist ein Design– bzw. Entwurfsmodell auf dem Symbol Level. Der Designprozeß besteht aus folgenden Schritten:

- der Auswahl eines Architekturparadigmas, beispielsweise eines objekt–orientierten oder funktionalen Ansatzes,
- der Spezifikation der Architektur,
- der Auswahl geeigneter Repräsentationstechniken,
- der Auswahl einer Softwareumgebung, d.h. einer Programmiersprache, eines Tools, Libraries etc. und
- der Implementierung.

Die Auswahlentscheidungen über die Architektur– und Repräsentationstechniken sind bereits angedeutet worden. Für die Implementierung des Systems ist ein objekt–orientierter Ansatz gewählt worden, der in bestimmten Bereichen auch durch funktionale Teile ergänzt worden ist (siehe 5.1.3). Damit kommen in diesem System sowohl die im vorangegangenem Kapitel beschriebenen wissensbasierten Methoden zum Einsatz als auch eher konventionelle Software–Techniken. Erst durch die Kombination dieser beiden Ansätze können alle Anforderungen an das System erfüllt werden.
Durch die Verwendung des Tools Delphi kann eine, den gängigen Softwarestandards entsprechende, benutzerfreundliche Oberfläche auf sehr einfache Weise implementiert werden.

Diese Softwareumgebung ist in diesem Fall auch geeignet, die im Rahmen des wissensbasierten Ansatzes gelernten Regeln zu repräsentieren. Erste Ideen, über eine PROLOG– bzw. C–Schnittstelle auf eine in PROLOG implementierte Wissensbasis (und Inferenzmaschine) zuzugreifen, wurden daher nicht weiter verfolgt. Mit Unterstützung des Tools Delphi können auch die für die Analyse erforderlichen Methoden umgesetzt werden.
In KADS wird unterschieden zwischen offenen und geschlossenen Software Environments (siehe [Schreiber et al, 1993], Seite 123). In offenen Environments besteht die Möglichkeit, die Menge der verfügbaren Methoden und Repräsentationen zu erweitern.
Bei dem Tool Delphi handelt es sich um eine offene Umgebung: dadurch, daß der Benutzer des Tools Zugriff auf Turbo bzw. Object Pascal hat, bestehen vielfältige Möglichkeiten zur Erweiterung der vorhandenen Methoden, Komponenten etc. Außerdem ist Delphi im Sinne der KADS–Einteilung eine sogenannte starke Umgebung, d.h. es wird eine große Anzahl von vordefinierten Techniken zur Verfügung gestellt. Das Tool vereinigt damit die Vorteile einer starken Umgebung mit der Flexibilität von offenen Umgebungen.
Dieses Kapitel teilt sich in zwei Abschnitte: die Spezifikation der Systemarchitektur (Abschnitt 5.1) und die Beschreibung des Designs der einzelnen Komponenten des Systems (Abschnitt 5.2).

5.1 Spezifikation der Systemarchitektur

Im Anschluß an die Auswahl eines Architekturparadigmas muß die Systemarchitektur genauer spezifiziert werden. Dazu werden in diesem Abschnitt zunächst eine Reihe von Anforderungen an das System CORA aufgestellt. Diese Anforderungen fließen in die Konzeption des Systemaufbaus ein: eine Beratungskomponente und ein Hilfesystem sowie die benutzerfreundliche grafische Oberfläche werden als zentrale Einheiten des Systems betrachtet (siehe 5.1.2). Der Aufbau des Systems ist im Rahmen dieser Arbeit bereits mehrfach angedeutet worden und wird daher hier nur kurz diskutiert. Schließlich geht dieser Abschnitt auf die zentralen Objekte des Systems ein, indem die Datenstrukturen für die Fall–Kontroll–Studiendaten vorgestellt werden.

5.1.1 Ergonomische Anforderungen an das System und ihre psychologischen Implikationen

Computerprogramme, insbesondere interaktive Assistentensysteme wirken auf allen Ebenen psychologischen Erlebens und Verhaltens. Daraus lassen sich entsprechende Forderungen an die Gestaltung und Strukturierung des Systems ableiten. Diese werden im folgenden kurz diskutiert.

Konsistenz: Konsistenz bedeutet, daß gleiche Vorgänge durch gleichartige Interaktionen bewirkt werden. Der Dialog sollte daher den Erwartungen der Benutzer entsprechen, die sie aus Erfahrungen mit Arbeitsabläufen mitbringen, und die sich während der Benutzung des Systems bilden. Uneinheitliches Dialogverhalten zwingt den Benutzer zu starker Anpassung an wechselhafte Durchführungsbedingungen seiner Arbeit und kann unnötige Belastungen mit sich bringen (siehe [DIN, 1984]). Sind Benutzerführung und Aussehen erwartungskonform, wird ein Programm eher akzeptiert als bei erwartungswidrigem Verhalten des Programms. Wiedererkennungsleistungen steigern die Akzeptanz bzw. die Affinität zum Programm.

Für das System CORA wird daher sowohl eine innere Konsistenz, also die Konsistenz innerhalb des Systems, als auch eine äußere Konsistenz, d.h. die Konsistenz in bezug auf andere Systeme, gefordert.

CORA läuft unter der grafischen Oberfläche Microsoft–Windows. Auch wenn einige Elemente von Windows nicht psychologischen Kriterien entsprechen, liegt es nahe, CORA den von Microsoft empfohlenen Standards anzupassen. Die standardisierten Windows-Anwendungen verleiten den Benutzer zum „intuitiven“ Vorgehen im Rahmen des Programms, d.h. die Anforderungen des Programms können so intuitiv erfaßt werden.

Adaptierbarkeit: Assistentensysteme werden von Benutzern mit unterschiedlichem Wissensstand bedient. Sie müssen daher in der Lage sein, einem ungeübten Benutzer möglichst einfach und eingängig genügend Informationen zur Verfügung zu stellen, so daß der Anwender entsprechende Entscheidungen fällen kann. Andererseits muß dem geübten Benutzer die Möglichkeit gegeben werden, das Programm schnell und ohne für ihn überflüssige Anweisungen bedienen zu können. Überflüssige Aktionen und redundante Mehrfachabfragen wirken sich demotivierend aus, verärgern den Anwender und erschweren die Akzeptanz des Programms. Ähnliches gilt für die Benutzerinformation: wichtig ist bei entsprechend breit gefächerter Benutzerqualifikation, daß sowohl ein wenig qualifizierter Benutzer zur Weiterarbeit motiviert wird als auch der routinierte Anwender. Zur Realisierung dieses Konzeptes ist es erforderlich, daß sich das System den unterschiedlichen Benutzern anpassen kann. So müssen kontext–sensitive Informationen und Hilfen unterdrückt bzw. angefordert werden (siehe Abschnitt 5.2.4).

Neben den beschriebenen kognitiven Prozessen sind auch emotionale Prozesse zu berücksichtigen. Daher müssen persönliche Vorlieben des Benutzers in das Programm einfließen können. In CORA sollte daher z.B. die Farbwahl für die Oberfläche vom Anwender beeinflußt werden können.

Visualisierung:

Psychologische Primärprozesse bei der Programmbenutzung sind Wahrnehmungs- und Informationsverarbeitungsprozesse. Aufmerksamkeit und simultane Informationsverarbeitungskapazitäten des Menschen sind limitiert. Entsprechend muß ein Programm die Aufmerksamkeit auf wesentliche Elemente fokussieren, z.B. durch symbolische und abstrahierte Darstellungen. Zu viele grafische Elemente steigern jedoch umgekehrt den Komplexitätsgrad der Wahrnehmungsvorlage, sie erschweren die Informationsverarbeitung. Daher ist es wichtig, einheitliche grafische Elemente mit Wiedererkennungswert in geringer Anzahl zu verwenden.

Ein wichtiger Aspekt ist hier die Gestaltung der Texte. Dies bezieht sich zum einen auf die Formulare des Programms, aber vor allem auch auf die Hilfetexte. Von Bedeutung ist hierbei, daß der Textumfang gering gehalten wird und insbesondere die Absätze kürzer sind als bei gedrucktem Text. Texthervorhebungen sollten sparsam eingesetzt werden. Es empfiehlt sich, nur zwei bis drei Schriftgrößen zu verwenden: einen größeren Schriftgrad, der dem aktuellen Fensterschlagwort zugeordnet wird, sowie einen kleineren Schriftgrad für den weiteren Text.

Die Aufgliederung des Textes in ein Schlagwort als Kurzfassung des Fensterinhaltes und den eigentlichen Text beruht auf lern- und gedächtnispsychologischen Überlegungen. Durch das Schlagwort werden bestimmte, inhaltsbezogene Assoziationen beim Leser geweckt, die die Aufmerksamkeit auf einen bestimmten Inhalt richten. Fehlerhafte Interpretationen des folgenden Textes können so reduziert werden. Aus gestaltpsychologischen Überlegungen werden in allen Bildschirminhalten zusammengehörige Elemente gruppiert und vom Rest separiert.

Dialogorientierte Benutzerführung: Die Benutzerführung geschieht im Rahmen des Assistentensystems CORA interaktiv. Dadurch soll die Benutzung des Programms erleichtert werden. Insbesondere ermöglicht es Anwendern mit unterschiedlichem statistischen Wissensstand, das Programm adäquat einzusetzen.

Zur Unterstützung der Benutzer wird der grobe Ablauf des Programms vom System geleitet. Damit werden unerfahrene Benutzer, die nicht mit dem genauen Ablauf der Analyse vertraut sind, unterstützt. Das System ist allerdings so flexibel, daß ein erfahrener Benutzer die Reihenfolge selbst bestimmen und auch ggf. Schritte auslassen kann.

Die Anforderungen an das System, die sich aus konstruktionsorientierter Sicht ergeben, sind die einfache Änderbarkeit und die rasche Erstellbarkeit, die durch die Verwendung des Tools Delphi erfüllt werden können (siehe Abschnitt 5.2).

Human Computer Interface → grafische Oberfläche		
Analysekomponente	**Hilfesystem**	**Beratungskomponente**
→ Schichtung → Aufbereitung → Analysen	→ Programmbenutzung → Analysekomponente → Beratungskomponente → stat. Hintergrund	→ Empfehlungen → Pilotstudie

Abbildung 5.1: Das System CORA

5.1.2 Aufbau des Systems

Zur Umsetzung der im vorangegangenen Abschnitt und im Anwendungsmodell aufgestellten Anforderungen an das System CORA ist der in Abbildung 5.1 gezeigte Aufbau des Systems konzipiert worden. Die Abbildung macht deutlich, daß den beratenden und unterstützenden Aspekten des Systems große Bedeutung zugemessen wird. Die Beratungskomponente des Systems, die auf den durch den wissensbasierten Ansatz gelernten Regeln basiert, enthält zusätzliche Informationen für den Anwender, die im Rahmen eines konventionellen Hilfesystems nicht zu finden sind.
Die Analysekomponente enthält alle im Anwendungsmodell aufgeführten Methoden, die im Rahmen der Kontingenztafelanalyse relevant sind.
Eine benutzerfreudliche grafische Oberfläche bietet dem Anwender einen einheitlichen, den Software–Standards entsprechenden Zugriff auf alle Systemkomponenten.

5.1.3 Die zentralen Objekte des Systems: Datenstrukturen für die Fall–Kontroll–Studiendaten

Das zentrale Konzept des Sachbereichs sind die Daten der durchgeführten Fall–Kontroll–Studie. Diese müssen für die Bearbeitung durch das System bestimmten Bedingungen genügen. Es können bis zu fünf Kovariablen in der Analyse berücksichtigt werden. Die Werte für diese Kovariablen können sowohl in Intervallen als auch als einzelne Werte angegeben werden.
Die Daten müssen in Form eines ASCII–Files mit der Endung „`.dat`“ vorliegen. Es ist das in Tabelle 5.1 dargestellte Format einzuhalten.
„ak“ ist ein Integer–Wert zwischen eins und fünf, der die Anzahl der vorhandenen Kovaria-

ak ai
$Name(k_1) \ldots Name(k_{ak})$
$Wert1.1(k_1)\ Wert1.2(k_1) \ldots\ Wert1.1(k_{ak})\ Wert1.2(k_{ak})\ a_1\ b_1\ c_1\ d_1$
$Wert2.1(k_1)\ Wert2.2(k_1) \ldots\ Wert2.1(k_{ak})\ Wert2.2(k_{ak})\ a_2\ b_2\ c_2\ d_2$
$Wert3.1(k_1)\ Wert3.2(k_1) \ldots\ Wert3.1(k_{ak})\ Wert3.2(k_{ak})\ a_3\ b_3\ c_3\ d_3$
...
$Wertn.1(k_1)\ Wertn.2(k_1) \ldots\ Wertn.1(k_{ak})\ Wertn.2(k_{ak})\ a_n\ b_n\ c_n\ d_n$

Tabelle 5.1: Format der ASCII–Datei

blen angibt. „ai" ist die Anzahl der Kovariablen, für die Intervalle angegeben werden. Diese Kovariablen müssen in der Datei zuerst aufgelistet werden. In der nächsten Zeile stehen die Namen der vorhandenen Kovariablen. Die Namen sind Strings, die durch ein Leerzeichen zu trennen sind. Die weiteren Zeilen der Datei bestehen aus den eigentlichen Daten: zuerst werden die Ausprägungen der Kovariablen angegeben; die letzten vier Werte der Zeile i geben die Anzahlen der exponierten Fälle (a_i), der nicht exponierten Fälle (b_i), der exponierten Kontrollen (c_i) und der nicht exponierten Kontrollen (d_i) in der Schicht i wieder. Für die Kovariablen, die nicht in Intervallen angegeben werden, fällt jeweils der zweite Wert weg, der sonst die obere Grenze darstellt. Sowohl die Werte für die Kovariablen als auch die Tafelwerte sind Fließkommazahlen vom Typ Single. Dies ist auch für die Tafelwerte, bei denen es sich eigentlich um Anzahlen handelt, notwendig, da Nullzellenkorrekturen durchgeführt werden können (z.B. eine Addition von 0.5 auf alle Zellenwerte).

Der Datensatz aus Tabelle 3.4 wird damit beispielsweise zu dem in Abbildung 5.2 dargestellten ASCII–File.

Im System werden die Daten in der nun beschriebenen Datenstruktur gehalten. Es wird ein Datentyp `TDaten` mit der in Tabelle 5.3 vorgestellten Deklaration entworfen. Die Methodendeklarationen sind in der Tabelle nicht aufgeführt.

Es gibt also sieben Felder, die die Eigenschaften des Datenobjekts enthalten. Durch die Methode `Einlesen`, die dem Objekttyp zugeordnet ist, werden die Werte aus der Datei in die Felder des Objekts eingelesen. Eine weitere Methode, `Schreibe_Memo`, schreibt die Eigenschaften des Datenobjekts in eine als Parameter übergebene Memokomponente. Diese Memokomponente (siehe 2.3) hat das gleiche Format wie die oben beschriebene ASCII–Datei. Auf diese Weise können die Daten wieder zurück in eine Datei geschrieben werden.

In der Anwendung werden eine Reihe von Instanzen des Typs `TDaten` erzeugt, nämlich

- die Daten für die Analysekomponente des Systems,

```
2 1
Alter Rauchen
25 29 0 0 1 25 16
25 29 1 4 1 37 118
30 34 0 0 0 13 175
30 34 1 9 12 20 215
35 39 0 0 3 8 153
35 39 1 4 30 18 177
40 44 0 1 10 4 165
40 44 1 5 55 5 197
45 49 0 3 20 2 155
45 49 1 3 73 3 146
```

Abbildung 5.2: ASCII–File zum Datensatz aus Tabelle 3.4

```
TDaten = class(TObject)
Dateiname : String;
AnzahlKov : Integer;
AnzahlSchichten : Integer;
Stichprobenumfang : Single;
Zelle : Array['a'..'d',1..100] of Single;
Kovariable : Array[1..5] of String;
KovWert : Array[1..100,1..5,1..2] of Single;
```

Abbildung 5.3: Der Datentyp `TDaten`

- die Daten für die Beratungskomponente, also der Pilotdatensatz und der Analysedatensatz sowie
- die Daten, die durch mögliche Schichtungen entstehen.

Alle Berechnungen der Punkt– und Varianzschätzer, der Homogenitäts– und Unabhängigkeitstests, die Schichtung und die Erzeugung der Pilotdaten greifen auf die Daten zu und bearbeiten sie. Es erscheint zunächst sinnvoll, die Prozeduren und Funktionen für die Berechnungen dem Objekt als Methoden zuzuordnen. Es stellte sich aber heraus, daß es übersichtlicher ist, diese Berechnungen funktional zu definieren, da auf diese Weise die Komplexität des Objekttyps `TDaten` überschaubar bleibt. Diese Funktionen sind nun in einer Reihe unterschiedlicher Units definiert. Das Datenobjekt wird als Parameter übergeben.

5.2 Design der Systemkomponenten

Die im Abschnitt 5.1.2 vorgestellten Komponenten des Systems werden im folgenden aus einem Implementationsblickwinkel betrachtet. Dabei werden vor allem die konzeptionellen Designaspekte betont, aber auch einige Implementierungsdetails herausgegriffen und genauer vorgestellt.

5.2.1 Design der Benutzerschnittstelle

Das Sytem CORA läuft, wie bereits mehrfach erwähnt, unter der grafischen Oberfläche MS Windows. Die Benutzerschnittstelle besteht aus einer Reihe von Formularen, die mithilfe des Tools Delphi (siehe 2.3) erstellt worden sind.

Das Hauptformular der Anwendung (siehe Abb. 5.4) setzt sich zusammen aus einer Titelleiste, einer Menüzeile, einer Leiste mit Schaltflächen für häufig verwendete Befehle des Menüs (Symbolleiste) und einer Statuszeile, in der kurze Informationen und Hilfen zum aktuellen Systemkontext angezeigt werden. Der Hauptteil in der Mitte des Formulars enthält keine für den Benutzer sichtbaren Komponenten. Hier werden die aufzurufenden (Unter–) Formulare plaziert.

Von zentraler Bedeutung für die Entwicklung der Benutzerschnittstelle ist das Hauptmenü der Anwendung. Wie auch aus der Abbildung 5.4 zu entnehmen ist, gibt es acht Menüpunkte.

Unter „Datei“ findet der Anwender die Befehle zum Erstellen, Öffnen, Drucken und Schließen der Dateien sowie zum Schließen der Anwendung.

Der Menüpunkt „Pilotstudie“ enthält die Befehle zum Erstellen, Erzeugen (durch das System), Öffnen, Speichern und Schließen der beiden Datensätze für die Pilotstudie. Außerdem

Abbildung 5.4: Das Hauptformular der Anwendung CORA

kann die Beratungskomponente des Systems mit dem Befehl „Empfehlung“ aufgerufen werden.

Die „Schichtung“ umfaßt die Befehle für den Aufruf der zur Verfügung stehenden Methoden und das Speichern und Schließen der erzeugten Datei.

Die beiden nächsten Menüpunkte gehören ebenfalls zur Analysekomponente. Unter „Tafeln“ finden sich drei Befehle, mit denen unterschiedliche Sichten auf die Daten aufgerufen werden können. Es handelt sich dabei um die einzelnen Kontingenztafeln sowie einen Überblick über die Nullzellen und die individuellen Odds Ratios.

Die wichtigsten Analyseschritte sind unter dem Menüpunkt „Analyse“ angeordnet: die „Homogenitätsanalyse“, die Schätzung für das „Gemeinsame Odds Ratio“ und der „Unabhängigkeitstest“. Mit dem vierten Befehl „Ergebnisprotokoll“ kann ein Protokoll der durchgeführten Analyseschritte angezeigt werden.

Der Punkt „Fenster“ enthält Befehle zur Anordnung und zum Schließen der geöffneten Fenster.

Unter „Optionen“ kann eine Standardanalyse eingestellt, eine Formatüberprüfung der ASCII–Datei durchgeführt und eine Farbwahl für die Oberfläche vorgenommen werden.

Der letzte Menüpunkt bietet die Möglichkeit, das Hilfesystem aufzurufen. Der Befehl „Suchen“ öffnet das Formular von „Windows Help“ für das Suchen eines Topics oder Stichwortes im Hilfesystem. Außerdem kann zum Inhaltsverzeichnis und zu zwei unterschiedlichen Einführungssequenzen, den sogenannten Tutorien (siehe 5.2.4), gesprungen werden. Der letzte Befehl dieses Menüpunktes ruft das Infoformular der Anwendung auf, also ein Formular, das die Copyright–Informationen und das Programmlogo anzeigt. Obwohl dieser Punkt genau genommen nicht zum Hilfesystem gehört, ist er dennoch in allen Windows–Anwendungen an dieser Stelle zu finden.

Die Reihenfolge der Menüpunkte legt dem Anwender eine mögliche Abfolge der Analyseschritte nahe. Es ist auch zu beachten, daß nicht zu jedem Zeitpunkt alle Menüpunkte bzw. Unterpunkte (Befehle) aufgerufen werden können. Sowohl die Menüpunkte als auch die Unterpunkte können durch das Setzen ihrer Eigenschaft „Enabled“ auf den Wert „False“ deaktiviert werden. Ist beispielsweise noch keine Datei geöffnet, so können die Menüpunkte der Analysekomponente, die Befehle zum Speichern etc. durch den Anwender nicht angeklickt werden bzw. die mit diesen Befehlen verbundenen Ereignisbehandlungsroutinen werden nicht aufgerufen.

Wird ein Befehl des Menüs mit den Pfeil–Tasten ausgewählt, so wird dieser in der Statuszeile kurz beschrieben.

Das Haupt– oder Rahmenformular enthält weitere für den Anwender nicht ständig sichtbare Komponenten. Sie werden bei Bedarf aufgerufen. Hierbei handelt es sich vor allem um ei-

Abbildung 5.5: Dialog zum Öffnen einer Datei

ne Reihe von Dialogen zum Öffnen, Speichern und Drucken von Dateien. Der Benutzer des Systems kann, wie bereits erwähnt, für die verschiedenen Komponenten des Systems unterschiedliche Datensätze verwenden. Mithilfe der gerade erwähnten Dialoge kann der Anwender diese Datensätze aus Dateien einlesen (Öffnen), einen Datensatz als Datei abspeichern oder ihn ausdrucken.

Bei diesen Dialogen (siehe z.B. Abbildung 5.5) handelt es sich um Komponenten aus der Delphi–Bibliothek. Sie entsprechen den Dialogen in anderen Windows–Applikationen. Durch das Setzen einiger Eigenschaftswerte können sie der Anwendung angepaßt werden. Es wird beispielsweise ein Filter definiert, so daß nur Dateien mit der Endung „`.dat`" geöffnet werden können.

Es gibt drei Formulare, die die unterschiedlichen Datensätze anzeigen: ein Formular, das den Datensatz der Analysekomponente enthält, ein Formular, das gegebenenfalls den geschichteten Datensatz zeigt und schließlich ein Formular mit den beiden Datensätzen für die Beratungskomponente. Werden die Datensätze geöffnet bzw. erzeugt, so wird das entsprechende Formular aufgerufen. Die Formulare bestehen lediglich aus einer Titelzeile, die den Dateinamen anzeigt und einer Memokomponente (siehe Abschnitt 2.3), die den Datensatz zeilenweise im oben beschriebenen Datei–Format darstellt (siehe Abb. 5.4).

In diesen Formularen können die Datensätze verändert werden, d.h. der Anwender kann die Dateien editieren und die geänderten Daten abspeichern. Dieser Vorgang kann auch ohne das Öffnen einer Datei durchgeführt werden und zwar durch die Befehle „Datei|Neu" und „Pilotstudie|Neu".

Die Datenformulare bleiben solange geöffnet, bis die entsprechende Datei geschlossen wird.

Werden weitere Formulare geöffnet, so ist das Datenformular als Symbol weiterhin vorhanden und kann wieder dargestellt werden.
Alle Formulare, die der Anwender im Rahmen der Beratungs– und Analysekomponenten aufrufen kann, werden im Hauptformular der Anwendung plaziert. Abhängig vom Typ der Formulare sind gegebenenfalls mehrere verschiedene Formulare gleichzeitig sichtbar. Außerdem gibt es sogenannte „Children–Formulare“, die selbst mehrmals erzeugt werden können, siehe 5.2.3. Die Oberflächen für die Beratungs– und Analysekomponente des Systems werden in den folgenden Abschnitten zusammen mit den zugundeliegenden Prozeduren und Funktionen vorgestellt.

5.2.2 Design der Beratungskomponente

Die Beratungskomponente des Systems kann in Verbindung mit der Durchführung einer Pilotstudie aufgerufen werden. Die Empfehlung eines Schätzers basiert auf zwei Datensätzen, dem Pilotstudiendatensatz und dem Analysedatensatz. Diese können mithilfe des Systems aus einem Gesamtdatensatz erzeugt werden.
Der Menüpunkt „Pilotstudie“ hat eine Reihe von Unterpunkten zum Öffnen, Erzeugen, Speichern und Schließen der beiden Datensätze. Zunächst müssen die Daten eingegeben, erzeugt oder geöffnet werden. Die Erzeugung der Datensätze wird nun genauer betrachtet. Nachdem eine Datei, die den gesamten Datensatz enthält, mithilfe einer entsprechenden Dialogkomponente geöffnet worden ist, ist der Anwender aufgefordert, die Methode für die Erzeugung der Daten anzugeben. Dazu wird ein Formular geöffnet (siehe Abbildung 5.6), in dem die beiden zur Verfügung stehenden Methoden ausgewählt werden können, also die einfache und die geschichtete Zufallsauswahl. Desweiteren kann abhängig von der Methodenauswahl die Anzahl der Stichproben für den Pilotdatensatz bzw. eine Prozentangabe für den Anteil der Pilotdaten einer Schicht an den Gesamtdaten der Schicht eingegeben werden.
Beide Eingaben des Benutzers werden über Auswahlfelder (siehe Abschnitt 2.3) ermöglicht. Für die Änderung der Voreinstellung „Geschichtete Zufallsauswahl“ kann die Liste mit beiden Methodennamen „aufgeklappt“ und das entsprechende Listenelement durch Anklicken ausgewählt werden. Abhängig von dieser Auswahl ändert sich die Liste der auszuwählenden Elemente für den Stichprobenumfang der Pilotstudie. Für die geschichtete Zufallsauswahl bietet die Liste die Werte 10%, 20%, . . . , 90% an. Im anderen Fall besteht die aufklappbare Liste aus den natürlichen Zahlen von eins bis zum Stichprobenumfang des angegebenen Gesamtdatensatzes. Im Unterschied zur Methodenliste ist hier das Eingeben eines ggf. nicht in der Liste befindlichen Wertes durch den Anwender möglich.
Durch Anklicken des OK–Schalters dieses Formulars wird abhängig von der Methodenauswahl die Prozedur `Erzeuge_Pilotdaten_Einfach` bzw. `Erzeuge_Pilotdaten_Geschichtet`

Abbildung 5.6: Das Formular „DlgStichprobenmodus“

aufgerufen. Als Parameter erhalten beide das Objekt `GesamtDaten` vom Typ `TDaten`, in das die Werte aus der angegebenen Gesamtdatei eingelesen werden. Der zweite Parameter ist für die erste Prozedur der ausgewählte Stichprobenumfang, also ein Integer–Wert, für die zweite Prozedur wird ein Single–Wert übergeben. Dieser wird aus der Eingabe des Anwenders abgeleitet. So wird z.B. der Wert 10%, der in der Auswahlliste noch als String vorliegt, in den Single–Wert 0.10 umgewandelt. Die beiden Algorithmen für die Prozeduren werden in den Abbildungen 5.7 und 5.8 vorgestellt.

Der zweite Schritt in der Erzeugung der Datensätze wird durch den Aufruf der Prozedur `Entferne_Nullzeilen` durchgeführt, die die unbesetzten Zeilen einer Datei streicht.

Die so erzeugten Datensätze liegen in den Objekten `AnalyseDaten` und `PilotDaten` vor. Mithilfe der Methode `Schreibe_Memo`, die innerhalb des Objekttyps `TDaten` deklariert ist, werden die Datensätze als Zeilen von Strings umgeformt und in die entsprechenden Memofelder, die als Parameter übergeben werden, geschrieben. Damit ist die Erzeugung der Daten abgeschlossen. Für den nächsten Schritt ist das Vorhandensein dieser Datensätze unbedingt erforderlich. Sollen diese nicht durch das System erzeugt werden, können auch bestehende Dateien geöffnet oder Datensätze innerhalb des Systems neu geschrieben werden.

Das zentrale Element der Beratungskomponente ist die Empfehlung eines Punktschätzers für das gemeinsame Odds Ratio. Diese kann nach dem Erstellen oder Öffnen der erforderlichen Datensätze im Menüpunkt „Pilotstudie“ mit dem Befehl „Empfehlung“ aufgerufen werden.

Input: `Daten, Gesamtanzahl der Beobachtungen N,`
`Stichprobenumfang der Pilotdaten n`

Output: `Pilotdaten, Analysedaten`

Ablauf:

```
Analysedaten:= Daten;
Pilotdaten:= leerer Datensatz;

for i:=0 to (n - 1) do
begin
   Generiere eine ganzzahlige Zufallszahl z aus [1,N-i];
   Streiche die Beobachtung z aus den Analysedaten;
   Füge die Beobachtung z in den Pilotdatensatz ein;
end.
```

Abbildung 5.7: Die einfache Zufallsauswahl

Die Empfehlung wird in einem dreiseitigen Formular angezeigt. Auf der ersten Seite (siehe Abbildung 5.9) erhält der Anwender die Information, welcher Punktschätzer und welche Biaskorrektur durch das System für die Analyse der Daten in der Analysekomponente vorgeschlagen werden, sowie Informationen über den Eignungsgrad des Schätzers (siehe 4.2.2) und die Nummer der Regel, aus der die Empfehlung abgeleitet wird.

Die zweite Formularseite (siehe Abbildung 5.10) zeigt die Eigenschaften der Analysedaten an, nämlich die Anzahl der Tafeln, die durchschnittliche Fallanzahl, die Balanciertheit der Fallanzahlen, das durchschnittliche Verhältnis von Fällen und Kontrollen sowie die Balanciertheit dieses Verhältnisses. Hier werden die konkreten Werte für die Eigenschaften und ihre Bewertungen, d.h. die kategorisierten Eigenschaften angezeigt.

Die letzte Formularseite führt die Eigenschaften auf, die aus den Pilotdaten berechnet werden, also die geschätzte Größe des gemeinsamen Odds Ratios, die durchschnittliche Expositionswahrscheinlichkeit und ihre Balanciertheit sowie die Differenz zwischen der maximalen und minimalen Expositionswahrscheinlichkeit. Auch diese Eigenschaften werden bewertet.

Die Eigenschaften, die der Empfehlung zugrunde liegen, also als Prämissen in der angewendeten Regel vorkommen, werden auf den beiden gerade beschriebenen Formularseiten markiert.

Im folgenden wird nun die Implementierung der Beratungskomponente genauer betrachtet.

Input: Daten, Anzahl der Schichten K

```
      Beobachtungsanzahl in der k--ten Schicht N[k], k=1,...,K
      Anteil der Beobachtungen für die Pilotdaten p
```

Output: Pilotdaten, Analysedaten

```
Analysedaten:= Daten;
Pilotdaten:= leerer Datensatz;

for k:=1 to K do
begin
   B[k]:=N[k];
   n[k]:=N[k]*p;                   { neue Beobachtungsanzahl für die Pilotdaten}
end;
```

Ablauf:

```
for k:=1 to K do
begin
   for i:=0 to (n[k]-1) do
   begin
     Generiere eine ganzzahlige Zufallszahl z aus [1,B[k]];
     Streiche die Beobachtung z aus den Analysedaten;
     Füge die Beobachtung z in den Pilotdatensatz ein;
     B[k]:=B[k]-1;                { neue Beobachtungsanzahl der Analysedaten }
   end;
end.
```

Abbildung 5.8: Die geschichtete Zufallsauswahl

Abbildung 5.9: Erste Seite des Empfehlungsformulars

Abbildung 5.10: Zweite Seite des Empfehlungsformulars

Zur Generierung der Empfehlung sind folgende Schritte erforderlich:

- die Berechnung der Eigenschaften der Analyse– und der Pilotdaten,
- die Bewertung dieser Eigenschaften und
- die Auswahl einer passenden Regel aus einer integrierten Regelmenge.

Für den ersten Schritt gibt es einfache Funktionen, wie z.B. die Funktion `AvFallanzahl(Daten : TDaten) : Single`, die aus dem übergebenen Datenobjekt die durchschnittliche Fallanzahl berechnet. Etwas komplexer ist die Berechnung der Gini–Koeffizienten für die Balanciertheitseigenschaften. Als Ergebniswerte dieser Funktionen werden Werte zwischen 0 und 1 geliefert. Diese Eigenschaften werden nicht als Felder des Datentyps `TDaten` implementiert, da sie nur an dieser Stelle benötigt werden. Zur Schätzung des gemeinsamen Odds Ratios der Pilotdaten wird der Mantel–Haenszel Schätzer verwendet, d.h. auf die gegebenenfalls nullzellenkorrigierten Pilotdaten wird die Funktion `Berechne_MantelHaenszel(Daten : TDaten) : Single` aus der Analysekomponente des Systems angewendet.
Die mithilfe dieser Funktionen berechneten Werte für die Dateneigenschaften der Analyse– und der Pilotdaten werden entsprechenden Feldern zugeordnet, z.B.:

```
Fallanzahl.Text := FloatToStr(AvFallanzahl(AnalyseDaten)).
```

Damit wird der in einen String umgewandelte Rückgabewert der Funktion `AvFallanzahl` vom Typ Float als Text des Feldes Fallanzahl dargestellt.
Für die Bewertungen werden die Werte der gerade erwähnten Felder herangezogen. Sobald sich der Wert eines Feldes ändert (d.h. das Ereignis OnChange eintritt), wird eine entsprechende Prozedur aufgerufen, die den Wert beurteilt und einem weiteren Feld zuweist, z.B.:

```
procedure TDlgEmpfehlung.FallanzahlChange(Sender : TObjekt);
begin
     if (StrToFloat(Fallanzahl.Text) <= 5)
     then KFallanzahl.Text:='klein'
     else if (StrToFloat(Fallanzahl.Text) <= 20)
          then KFallanzahl.Text:='mittel'
          else if (StrToFloat(Fallanzahl.Text) <= 100)
              then KFallanzahl.Text:='groß'
              else KFallanzahl.Text:='sehr groß';
end;
```

Die kategorisierte Eigenschaft für die Fallanzahl steht damit im Textwert des Feldes KFallanzahl. Völlig analog werden auch alle weiteren Eigenschaften in die Kategorien eingeteilt. An dieser Stelle soll noch einmal deutlich gemacht werden, daß die Einteilung in die Kategorien hier der Einteilung entsprechen muß, die im Rahmen der Auswertung der Simulationsstudien modelliert wurde (siehe Tabelle 4.1).

Damit sind die Grundlagen für den dritten Schritt der Generierung von Empfehlungen geschaffen. Die Einkodierung der Regelmenge erfolgt in der Prozedur `Empfehlung (Sender : TObjekt)`, die nach der Berechnung der Eigenschaften durch den Befehl „Empfehlung“ im Hauptmenü aufgerufen wird. Die Prozedur setzt den in 4.10 aufgestellten Entscheidungsbaum in Form von Produktionsregeln durch geschachtelte if–Abfragen um. In den if–Schleifen werden die Feldwerte für die kategorisierten Dateneigenschaften entsprechend der Prämissen der Regeln in der festgelegten Reihenfolge abgefragt. Jeder Bedingungsteil fragt eine Dateneigenschaft, also eine Prämisse ab:

```
if (KFallanzahl = 'klein')
then if (KTafelanzahl = 'mittel')
      then if ...
else if (KFallanzahl = 'mittel')
      then ...
```

Sind alle Prämissen einer Regel abgefragt, so wird im zugehörigen then–Teil die Konklusion der Regel repräsentiert. Die Konklusionen enthalten folgende Informationen: den zu empfehlenden Schätzer bestehend aus dem Basis–Schätzer und gegebenenfalls den Modifikationen sowie den Eignungsgrad des Schätzers. Zusätzlich wird jeder Regel eine Nummer zugeordnet, damit diese eindeutig identifiziert werden kann.

Diese Informationen werden den Textwerten der entsprechenden Felder der ersten Seite des Empfehlungsformulars zugewiesen. Es handelt sich dabei um die Felder „Punktempfehlung“, „Biaskorrekturempfehlung“, „Eignung“ und „Regelnummer“. Zur Verdeutlichung betrachte man folgendes Beispiel:

Die Konklusion der ersten Regel (`schlechte_empfehlung(S,w_jk)`) wird repräsentiert durch

```
... (Abfrage der Prämissen)
then begin
       Punktempfehlung.Text := 'Woolf';
```

```
        Biaskorrekturempfehlung.Text := 'Jackknife II';
        Eignung.Text := 'schlecht';
        Regelnummer.Text := 'R1';
       end
```

Desweiteren wird es als sinnvoll erachtet, dem Benutzer auch die Prämissen der angewendeten Regel anzuzeigen. Daher sind die entsprechenden Dateneigenschaften im Empfehlungsformular gekennzeichnet. Es gibt zu jeder Dateneigenschaft ein Markierungsfeld, das angekreuzt werden kann. Diese Markierung wird ebenfalls in den then–Teilen der if–Abfragen festgelegt. Nach jeder erfolgreichen Abfrage wird das zugehörige Feld angekreuzt, d.h. der Wert des Markierungsfeldes wird auf „True“ gesetzt:

```
if (Fallanzahl.Text = 'klein')
then begin
       CheckedFallanzahl := True;
       ...
```

Damit sind die Informationen, die aus den Regeln zu entnehmen sind, für den Benutzer weitestgehend transparent. Es ist zu überlegen, ob weitere Informationen, wie z.B. die Anzahl der von der Regel abgedeckten Prämissen, sichtbar gemacht werden sollen.

In 4.10 wurde betont, daß es Situationen gibt, in denen mehr als eine Regel anwendbar ist. Daher sollte eine bestimmte Abfragereihenfolge nicht nur für die Prämissen, sondern auch für die Regeln eingehalten werden. Das heißt die Suche im Entscheidungsbaum wird gestoppt, falls eine erste Regel angewendet werden kann. Dieses Verhalten wird in den if–Abfragen durch die Turbo Pascal Prozedur `Exit` erreicht, die den Aufruf einer Prozedur stoppt. `Exit` wird in den then–Teilen der Prozedur `Empfehlung` aufgerufen, in denen eine Konklusion einer Regel repräsentiert wird.

Für die Codierung der in 4.7 vorgestellten 24 Regeln, die 87 Prämissen enthalten, werden 35 if–Abfragen benötigt, die eine Verschachtelungstiefe von bis zu fünf besitzen. Im folgenden wird ein Ausschnitt der Prozedur `Empfehlung` dargestellt:

```
... if ( KFallanzahl.Text = 'sehr groß')
then begin
       RelAvFall.Checked:=True;
       if (( KExpos.Text = 'zentriert') and
        ( KDifferenz.Text = 'klein'))
       then begin
             RelExpo.Checked:=True;
```

```
RelDiff.Checked:=True;
if ( KTafelanzahl.Text = 'klein')
then begin
       RelTafel.Checked:=True;
       Punktempfehlung.Text := 'Mantel Haenszel';
       Biaskorrekturempfehlung.Text := 'Jackknife II';
       Eignung.Text := '-';
       Regelnummer.Text := 'R7';
       Exit;
      end
else                ...
```

5.2.3 Design der Analysekomponente

Da die Implementierung dieser Komponente bis auf die Schichtung nicht vollständig zu der vorliegenden Diplomarbeit gehört und größtenteils von einer Statistikerin durchgeführt worden ist, konzentriert sich dieser Abschnitt auf die Schichtung und erwähnt die Homogenitätsanalyse, die Schätzung des gemeinsamen Odds Ratios, der individuellen Odds Ratios mit der Aufbereitung der Daten in Form von Tafeln und den Unabhängigkeitstest nur kurz.
Es gibt eine Hauptdatei, die als Grundlage für die Analysekomponente dient. Da die Schichtung einen gegebenen Datensatz verändert und nicht die Datei selbst, kann der erzeugte Datensatz als Datei abgespeichert werden.
Die Elemente der Analysekomponente können durch die Befehle, die unter den Menüpunkten „Schichtung“, „Tafeln“ und „Analyse“ zu finden sind, aufgerufen werden.
Der Menüpunkt „Schichtung“ enthält die Unterpunkte „Mit Intervallbildung“ und „Ohne Intervallbildung“ sowie Befehle zum Speichern und Schließen des geschichteten Datensatzes. Die mit den beiden ersten Befehlen verknüpften Aktionen werden nun beschrieben.
Wie bereits in 3.2 erwähnt wurde, stellt das System zwei Methoden zur Schichtung der Daten bereit: eine Schichtung durch das Weglassen von Kovariablen und eine Schichtung durch das Bilden neuer (größerer) Intervalle für die Kovariablen.
Zunächst wird die erste Methode näher betrachtet. Durch den Aufruf des Befehls „Ohne Intervallbildung“ wird das Formular „DlgSchichtung“ angezeigt (siehe Abbildung 5.11). Dieses Formular ist mithilfe einer Formularschablone von Delphi, der Schablone „Auswahldialog“ (siehe 2.3), entworfen worden. Die Schablone besteht aus zwei Listen: die links stehende Liste enthält die (noch) zur Auswahl stehenden Elemente; die Liste auf der rechten Seite zeigt die ausgewählten Elemente an. Es gibt vier Schalter ($\boxed{>}$, $\boxed{\gg}$, $\boxed{\ll}$ und $\boxed{<}$), mit denen ein markiertes Element bzw. alle Einträge in die entsprechende Richtung verschoben werden

Abbildung 5.11: Formular „DlgSchichtung“

können. Zusätzlich enthält die Schablone die drei Standardschalter „OK“, „Abbruch“ und „Hilfe“, die auf allen Formularen plaziert werden, die eine Eingabe des Benutzers erfordern. Diese Formularschablone wird komplett übernommen; lediglich einige Beschriftungen werden hinzugefügt bzw. geändert. Beim Aufruf des Formulars wird die i–te Zeile der links plazierten Liste mit dem Namen der i–ten Kovariable aus dem zugrundeliegenden Datensatz (`Daten.Kovariable[i]`) initialisiert (i = 1,…, `Daten.AnzahlKov`). Die rechts stehende Liste ist zu diesem Zeitpunkt leer. Der Anwender kann die Kovariablen, die im Rahmen der Schichtung berücksichtigt werden sollen, also die „Schichtungsvariablen“, von der linken Seite in die rechte Liste schieben. Zu Korrekturzwecken können Elemente aus der rechten Liste auch wieder entfernt, also zurückgeschoben werden.

Die Auswahl wird mithilfe der „OK“–Schaltfläche bestätigt. Die Ereignisbehandlungsroutine „`OkButton.OnClick`“ ruft die Funktion „`Schichtung`“ auf. Dieser Funktion werden folgende Parameter übergeben:

- das Datenobjekt vom Typ `TDaten`, das für die Analysekomponente verwendet wird, und
- ein Array von Typ `TSchichtungsvariablen (Array[1..5] of Boolean)`, in dem die Positionen der ausgewählten Kovariablen (rechte Liste) mit „True“ gekennzeichnet sind.

Vor dem Aufruf der Schichtung wird dieses Array von der erwähnten Ereignisbehandlungsroutine `OkButton.OnClick` erzeugt.

Input: Daten, Positionen der zu berücksichtigenden Koariablen

Output: geschichtete Daten

Ablauf:

```
Erzeuge einen Kovariablendatensatz (ohne Beobachtungen);
Streiche die doppelten Zeilen in diesem Datensatz,
die durch das Weglassen von Kovariablen entstehen können;

for i:=0 to Anzahl der Zeilen im Kovariablendatensatz do
begin
   for j:=1 to Anzahl der Zeilen im Ausgangsdatensatz do
   begin
     Suche alle Zeilen aus dem (ungeschichteten) Datensatz, die die
     gleichen Ausprägungen für alle betrachteten Kovariablen
     haben, wie die aktuelle Zeile des Kovariablendatensatzes
   end;
   Addiere die Anzahlen der Beobachtungen der gefundenen Zeilen und
   schreibe diese in die aktuelle Zeile des Kovariablendatensatzes;
end.
```

Abbildung 5.12: Algorithmus zur `Schichtung`

Der Rückgabewert der Funktion ist ein neues Datenobjekt von Typ `TDaten`, das den geschichteten Datensatz enthält. Die Funktion `Schichtung` ist relativ komplex; der Algorithmus wird in Abbildung 5.12 grob skizziert.

Damit ist die Schichtung durch die erste Methode abgeschlossen. Wählt der Anwender die zweite Methode durch den Befehl „Mit Intervallbildung“ im Hauptmenü aus, so wird das Formular „DlgIntervallSchichtung“ aufgrufen. Das Formular ist in Abbildung 5.13 dargestellt. Es teilt sich in vier wesentliche Teile, die durch Gruppierungselemente zusammengefaßt sind.

Die Gruppierung mit der Überschrift „Kovariable“ (siehe Abbildung 5.13 links oben) stellt Informationen über die gerade bearbeitete Variable bereit, nämlich den Namen sowie den kleinsten und größten Wert der Ausprägungen dieser Variable. Die Maxima und Minima der Kovariablen aus dem Datensatz werden mithilfe einfacher Funktionen berechnet. Das Feld, das den Namen der Kovariable anzeigt, ist ein Auswahlfeld, in dem alle Namen der im

Abbildung 5.13: Das Formular „DlgIntervallSchichtung“

Datensatz vorhandenen Kovariablen stehen. Der Anwender sucht hier die gewünschte Variable aus. Abhängig von dieser Auswahl ändern sich die Werte der anderen Felder.

Rechts neben dieser Gruppierung werden die in dem Datensatz vorhandenen Intervalle für die ausgewählte Kovariable dargestellt.

In der Gruppierung „Eingabe der Intervalle“ gibt der Benutzer die unteren und oberen Grenzen für die Intervalle an. Jedes Intervall muß durch Anklicken des „Berechnen“–Schalters bestätigt werden. Die Ereignisbehandlung für das Anklicken dieses Schalters besteht aus der Berechnung der Werte, die im vierten Gruppierungsfeld angezeigt werden, nämlich den neu gebildeten Intervallen, der Berechnung der Parameter der Funktion `Intervallschichtung` und ihrem Aufruf sowie dem Löschen der eingegebenen Grenzen. Die Funktion `IntervallSchichtung` wird also für jedes zu bildende Intervall aufgerufen. Sie benötigt folgende Parameter: das Datenobjekt, ein Array mit den Positionen der zu berücksichtigenden Variablen (analog zur Funktion `Schichtung`), die Position der Kovariablen, für die Intervalle angegeben werden, und einen Parameter, der angibt, an welcher Zeilenposition die neuen Schichten im Datensatz angefügt werden müssen. Die Zeilenposition wird durch die aktuelle Anzahl der Schichten des Output–Datenobjekts dieser Funktion bestimmt. Bei jedem Aufruf der Funktion werden also neue Schichten an den Datensatz angefügt. Dieser Algorithmus wird in Abbildung 5.14 skizziert.

Ist die Schichtung der Daten abgeschlossen, unabhängig davon, ob dies mit oder ohne Bildung von Intervallen erfolgte, wird der erzeugte Datensatz in einem ähnlichen Formular wie der Ausgangsdatensatz angezeigt. Dieses Formular enthält ebenfalls nur eine Memokomponente und zeigt den Datensatz im File–Format an. Mit den weiteren Befehlen im Menüpunkt „`Schichtung`“ kann der Datensatz als ASCII–File gespeichert und geschlossen werden.

Es ist zu beachten, daß das Ausgangsdatenobjekt durch die Schichtung nicht verändert wird. Das heißt also, daß der Analysekomponente auch nach einer Schichtung noch der ungeschichtete Ausgangsdatensatz zugrunde liegt. Eine erneute Schichtung oder auch die Analyse greifen damit auf den ursprüngliche Datensatz zurück. Einerseits hat dies den Vorteil, daß bei einer den Anwender nicht zufriedenstellenden Schichtung der Ausgangsdatensatz nicht neu eingelesen werden muß. Andererseits muß der Anwender so aber den geschichteten Datensatz für die Analysekomponente öffnen, sofern er mit diesen Daten weiterarbeiten möchte. Das heißt beispielsweise für eine Kombination der beiden Schichtungsmethoden — für eine Kovariable werden Intervalle angegeben, gleichzeitig sollen andere Kovariablen nicht mehr berücksichtigt werden — muß der Anwender zunächst einen Datensatz mit dem Befehl „Datei|Öffnen“ für die Analysekomponente einlesen, eine Schichtung durchführen, die Daten mit „Schichtung|Speichern Unter“ sichern, diese Daten wieder einlesen („Datei|Öffnen“) und die zweite Schichtung durchführen. Erste Erfahrungen aus der Anwendung des Systems wer-

Input: Daten, Position der betrachteten Kovariable,
untere Intervallgrenze, obere Intervallgrenze,
Positionen der insgesamt betrachteten Kovariablen, Anfügeposition

Output: geschichtete Daten

Ablauf:

```
for K:=0 to Anzahl der Zeilen im Datensatz do
begin
   Suche alle Zeilen, für die die Ausprägungen der betrachteten
   Kovariable innerhalb des angegebenen Intervalls liegt;
   Füge diese Zeile in den geschichteten Datensatz an der durch die
   Anfügeposition bestimmte Stelle ein;
   Schichte diesen neuen Datensatz mithilfe der Funktion Schichtung;
end.
```

Abbildung 5.14: Algorithmus zur `IntervallSchichtung`

den zeigen, ob diese Vorgehensweise sinnvoll ist oder besser das Ausgangsdatenobjekt — nicht die Datei — direkt überschrieben wird.

Die weiteren Elemente der Analysekomponente werden im folgenden überblickartig vorgestellt.

Bevor man die Informationen aus den Daten zusammenfaßt und analysiert, können die einzelnen Tafeln untersucht werden. Unter dem Menüpunkt „Tafeln“ gibt es die Unterpunkte „Einzelne Tafeln“, „Odds Ratios“ und „Nullzellen“, die verschiedene Sichten auf die Daten anbieten.

Mit dem ersten Befehl wird ein Formular vom Typ `TTafeln` (siehe Abbildung 5.15) aufgerufen, das die einzelnen Schichten des Datensatzes in Form von 2x2–Kontingenztafeln darstellt. Es kann jeweils eine Tafel angesehen werden. Da es sich bei diesem Formular um ein sogenanntes „Children–Formular“ handelt, können mehrere Formulare dieses Typs erzeugt und geöffnet werden. Somit können durch den Aufruf mehrerer Formulare eine Reihe von Tafeln gleichzeitig betrachtet werden. Die Anordung dieser Formulare kann durch den Anwender per Hand oder mithilfe der Befehle aus dem Menüpunkt „Fenster“ erfolgen. Welche Tafel im Formular angezeigt werden soll, wird durch den Anwender im Feld „Tafelnummer“ angegeben.

Die Formulare vom Typ `TTafeln` sind zweiseitige Formulare. Die zweite Seite zeigt die zur

Abbildung 5.15: Formular „Tafeln“

Tafel gehörenden Schätzungen des individuellen Odds Ratios, der Varianz und eines Konfidenzintervalls an. Einige Optionen für die Berechnungen stehen über Auswahlfelder zur Verfügung.

Die Formulare „Odds Ratios“ und „Nullzellen“ geben einen Überblick über alle individuellen Odds Ratios des Datensatzes und über vorhandene Nullzellen, also unbesetzte Felder in den Tafeln.

Die drei wesentlichen Analyseschritte, die Homogenitätsanalyse, die Schätzung des gemeinsamen Odds Ratios und der Unabhängigkeitstest, sind in ihrer Struktur völlig gleich aufgebaut. Wählt der Anwender einen entsprechenden Befehl aus dem Menüpunkt „Analyse“ aus, so erscheint ein folgendermaßen konzipiertes Formular: Es gibt zwei Gruppierungselemente, die die Formulare in unterschiedliche Bereiche aufteilen. In der oberen Gruppierung wird die Prozedur für den jeweiligen Analyseschritt ausgewählt, d.h. im Homogenitätsformular wird die Methode für die Homogenitätsanalyse bestimmt, im Formular für die Schätzung des gemeinsamen Odds Ratios werden der Punktschätzer und zugehörige Biaskorrekturen (Modifikationen), ein geeigneter Varianzschätzer, die Methode zur Berechnung von Konfidenzintervallen und die Überdeckungswahrscheinlichkeit ausgewählt, im Formular für den Unabhängigkeitstest wird die Testprozedur und das gwünschte Signifikanzniveau durch den Anwender angegeben.

Die Art der Auswahl ist in allen drei Formularen identisch: die Namen der zur Verfügung stehenden Prozeduren bzw. die Werte für die Niveaus stehen in Form von aufklappbaren Listen zur Auswahl.

Im Rahmen der Homogenitätsanalyse gibt es fünf Testmethoden. Dazu gehören der nicht–iterative Woolf–Test (siehe [Woolf, 1955]), der Breslow–Day–Test (siehe [Breslow, Day, 1980]), der MW–Test und der MW–Score–Test (siehe [Strugholtz, 1993]) sowie

Abbildung 5.16: Formular „Odds Ratios“

Abbildung 5.17: Formular „Nullzellen“

der approximative Tarone–Test (siehe [Tarone, 1985]). Die Punktschätzung des gemeinsamen Odds Ratios kann innerhalb des Systems mithilfe des Woolf– sowie des Mantel–Haenszel–Schätzers und den gejackkniften Versionen dieser Schätzer durchgeführt werden (siehe 3.3.1). Insgesamt ergeben sich damit also sechs verschiedene Punktschätzer, zwischen denen der Benutzer wählen kann. Für die Varianzschätzung des Mantel–Haenszel–Punktschätzers stehen die Jackknife–Varianz erster und zweiter Art (siehe [Pigeot, 1989]), die Breslow–, Breslow–Liang–, die Hauck– und die Robins–Breslow–Greenland–Varianz sowie die symmetrischen Varianzen der letzten vier Versionen zur Auswahl (siehe [Hauck, 1987]). Für den Woolf–Schätzer können die Jackknife–Varianzen erster und zweiter Art (siehe [Pigeot, 1989]) sowie die Woolf–Varianz (siehe [Gart, 1962]) berechnet werden. Die Konfidenzintervalle können symmetrisch oder schief (über die log–Transformation) berechnet werden.

Für den Unabhängigkeitstet bietet CORA den Benutzerinnen die folgenden vier Testmethoden an: den Mantel–Haenszel–Test mit und ohne Stetigkeitskorrektur, den Cochran–Test und den Woolf–Test (siehe [Hauck, 1989] und [Li et al, 1979] für Stetigkeitskorrekturen).

Unter dem gerade beschriebenen Gruppierungselement befinden sich die Schaltflächen dieses Formulars. Die „Berechnen“–Schaltfläche dient zur Bestätigung der oben getroffenen Auswahl und ruft die Funktionen zur Berechnung der Analyse– bzw. Testergebnisse auf. Diese werden im unteren Gruppierungsbereich des Formulars angezeigt. Die Abbildung 5.18 zeigt exemplarisch das Formular für die Schätzung des gemeinsamen Odds Ratios.

Mithilfe des Befehls „Optionen|Standardanalyse“ kann der Anwender einstellen, welche Prozeduren standardmäßig beim Öffnen eines Analysefensters angezeigt werden.

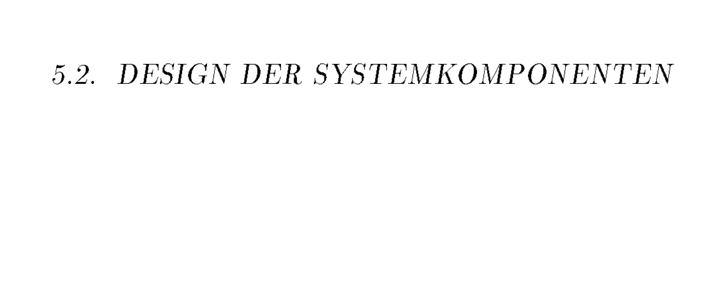


Abbildung 5.18: Das Formular zur Schätzung des gemeinsamen Odds Ratios

5.2.4 Design des Hilfesystems

Für die Implementierung des Hilfesystems wird das Programm „Windows Help“ (siehe 2.3.6) verwendet. Es gibt zwei Themendateien (.rtf–Files): eine Datei, die die statistischen und teilweise auch epidemiologischen Themen umfaßt, und eine Themendatei, in der das Programm und seine Benutzung erläutert wird. Die erste Datei ist nicht im Rahmen der Diplomarbeit erstellt worden. Sie wird von einer Statistikerin konzipiert und geschrieben. Daher wird nur die zweite Themendatei im folgenden kurz vorgestellt.

Die Themendatei enthält ca. 25 Topics. Das erste Topic ist ein gemeinsames Inhaltsverzeichnis, von dem aus Themen der beiden Themendateien erreichbar sind. Dieses Inhaltsverzeichnis kann über den Befehl „Hilfe|Inhalt“ des Hauptmenüs aufgerufen werden.

Ein einführendes Topic gibt einen Überblick über das System. Hier wird erläutert, wie das System aufgebaut ist, was es leistet und wie es eingesetzt werden kann. Ein weiteres Topic beschreibt den Ablauf der Analyse mit dem System CORA. Dabei werden alle Menübefehle des Hauptformulars kurz vorgestellt.

Die Beratungskomponente bedarf ebenfalls einer ausführlichen Erläuterung, die im Rahmen des Hauptformulars abgerufen werden kann. Auch die Analysekomponente wird hier detailliert vorgestellt.

Ein wichtiges Thema des Hilfesystems sind die Informationen über die Datensätze. Daher gibt es ein Topic, das erläutert, welche Systemkomponenten auf die verschiedenen Datensätze zurückgreifen. Von zentraler Bedeutung für den Anwender ist das Topic, das das Dateiformat beschreibt.

Alle Formulare des Systems CORA, die eine Eingabe durch den Anwender erfordern, besitzen eine Hilfe–Schaltfläche. Zu jedem dieser Formulare gibt es daher ein Hilfetopic, das die Komponenten des Formulars vorstellt und die Eingabemöglichkeiten für den Anwender in diesem Formular beschreibt. Außerdem kann von hieraus jeweils zu den relevanten statistischen Hilfethemen gesprungen werden. Dies sei am folgenden Beispiel erläutert:

> Das Formular „Homogenitätsanalyse“ besitzt einen Hilfeschalter, mit dem zu einem Topic gesprungen werden kann, das erläutert, wie die verschiedenen Prozeduren im Formular ausgewählt werden können, welche Funktionen die vorhandenen Schalter haben und was bestimmte Ergebniswerte aussagen (z.B. die Werte, die angezeigt werden, falls ein Nullzellenfehler aufgetreten ist).
>
> Von diesem Topic kann dann über Hotspots (siehe 2.3.6) zu den Themen gesprungen werden, die die Homogenitätsanalyse allgemein und die zur Verfügung stehenden Prozeduren erläutern. Dazu gehören beispielsweise auch Informationen

darüber, bei welchen Testergebnissen die Homogenitätshypothese abgelehnt werden sollte.

Über den Befehl „Hilfe|Suchen“ im Hauptmenü kann der Anwender auch direkt ein bestimmtes Hilfethema aufrufen. Alle Themen aus beiden Dateien sind über diese Funktion erreichbar.
Ein weiterer Menüpunkt aus dem Hilfemenü im Hauptformular ist die „Einführung in CORA“. Hier handelt es sich um eine Browse–Sequenz (siehe 2.3.6), die die ersten Topics, d.h. die oben erwähnte Einführung in das Programm und seine Benutzung, den Ablauf und die Informationen über die Datensätze und Dateien, umfaßt.
Eine ähnliche Einführung erhält der Anwender über den Befehl „Hilfe|Einführung in die KTA“. Diese Browse–Sequenz gibt dem Anwender einen kurzen Überblick über das Ziel und den Ablauf der Analyse geschichteter 2x2–Kontingenztafeln.

Kapitel 6

Bewertung und Einordnung des Systems CORA

In diesem Kapitel wird das vorgestellte System CORA anhand einer Reihe von Kriterien kritisch diskutiert und mit anderen Ansätzen verglichen. Zunächst werden die folgenden wesentlichen Konstruktionsentscheidungen untersucht:

- **Einsatzbereich des Systems:**
  Welcher Bereich der Statistik wird durch das System abgedeckt?
  Ist der Anwendungsbereich eingeschränkt?

- **Analysekomponente:**
  Hat das System eine Schnittstelle zu vorhandenen statistischen Softwarepaketen oder werden die angebotenen Methoden im System selbst implementiert?

- **Statistische Expertise:**
  Wie umfangreich ist die Problemlösungsexpertise des Systems?
  Basiert die Expertise auf datenabhängigem oder datenunabhängigem Wissen?

- **Dialogkomponente:**
  Ist der Dialog natürlichsprachlich, menügesteuert oder wird er über eine Kommandosprache oder eine grafische Benutzeroberfläche gesteuert?

- **Erklärungskomponente:**
  Sind die Entscheidungen des Systems für den Anwender transparent?
  Hat das System die „Fähigkeit“ zu unterrichten, d.h. kann der Anwender aus der Programmbenutzung eine Expertise erwerben?

- **Wissensakquisition:**
  Besitzt das System eine Komponente zur Wissensakquisition?
  Hat das System die Fähigkeit vom Anwender zu lernen?

Zu dem erstgenannten Kriterium ist anzumerken, daß sich das System CORA auf einen relativ kleinen Bereich der Statistik beschränkt, nämlich auf die Analyse geschichteter 2x2–Kontingenztafeln, der Anwendungsbereich jedoch nicht eingeschränkt wird. Wenn auch davon ausgegangen wird, daß das System typischerweise in der Epidemiologie eingesetzt wird, so sind doch ebenfalls andere, beispielsweise psychologische oder ökonomische Untersuchungen mithilfe des Systems möglich. Das Hilfesystem geht allerdings auf diese Anwendungsbereiche nicht in gleichem Maße ein wie auf epidemiologische Aspekte.
Fast alle bekannten statistischen Expertensysteme machen ebenfalls starke Einschränkungen bezüglich der zur Verfügung stehenden Methoden. Als eines der frühen Systeme soll hier beispielsweise REX (siehe [Gale et al, 1982]) genannt werden, das sich auf lineare Regression beschränkt, oder auch das System ESTES[1] (siehe [Hietala, 1986]), mit dem Zeitreihen analysiert werden können.
Bezüglich des zweiten Kriteriums, der Analysekomponente, unterscheidet sich CORA allerdings von einer Reihe der bekannten Systeme. GLIMPSE (siehe [Nelder, 1987]) baut beispielsweise auf dem Softwarepaket GLIM[2] auf, das einen Anwender bei der Bildung von Modellen für seine Daten unterstützt. Dem Benutzer von GLIMPSE wird der Zugriff auf alle Tools von GLIM ermöglicht. Dabei wird eine „High Level Command Language“ zur Formulierung von Aufgaben angeboten, die dann durch einen Preprozessor in GLIM–Anweisungen übersetzt werden. Der Output von GLIM wird für den Benutzer aufbereitet. Der Vorteil dieser höheren Kommandosprache liegt in den Hilfestellungen für den Anwender. In GLIMPSE wird diese Unterstützung auf drei Ebenen zur Verfügung gestellt:

Reminder Mode: In diesem Modus wird lediglich die Syntax der Anweisungen angezeigt. Er eignet sich damit für erfahrene Benutzer.

Prompting Mode: Hier wird eine Liste von verfügbaren Anweisungen angeboten.

Handholding Mode: Die schrittweise Entwicklung von Anweisungen wird ermöglicht, ohne die zugrundeliegende Kommandosprache zu kennen.

Dieses System basiert also auf der Idee, die komplexe Sprache eines statistischen Auswertungssystems für den Benutzer leichter handhabbar zu machen. Statistik–Kenntnisse werden vom Anwender aber in gleichem Maß erwartet wie für das zugrundeliegende System GLIM.

[1] Expert System for Time Series

[2] Generalized Linear Interactive Modelling

Die Autoren selbst sehen ihr System als „Knowledge–Based Front End“, nicht als Expertensystem.
Einen Schritt weiter gehen Schnittstellensysteme, die die semantische Korrektheit der Anweisungen überprüfen (siehe beispielsweise [Jida et al, 1986]). Andere Ansätze, wie z.B. SETUP (siehe [Naeve et al, 1987]) oder EXPRESS (siehe [Carlsen et al, 1986]), die ebenfalls auf statistischen Softwarepaketen aufbauen, setzen ihr Ziel noch etwas höher an. Der SETUP–Ansatz, dem das System P–STAT zugrunde liegt, will zusätzlich statistische Expertise in das System einbringen. Die Autoren halten daher u.a. folgende Leistungsmerkmale fest (siehe [Naeve et al, 1987], Seite 5):

> „... [Für das System werden eine] Auswahlunterstützung von P–STAT–Komponenten und Parameterempfehlungen aufgrund der Datensituation [gefordert]. Gedacht ist an eine Liste kontextabhängiger geeigneter Methoden, die jeweils durch Modell– und Datensituation bestimmt wird [...] und dem Benutzer zur Auswahl zur Verfügung steht. [...] Die Modellierung soll auch durch Empfehlungen zum weiteren Vorgehen unterstützt werden, wie z.B. eine Residualanalyse zur Modellevaluation, die wiederum in eine Transformationsempfehlung münden kann.“

Damit wird sowohl eine Parallele als auch ein wesentlicher Unterschied zum System CORA deutlich: CORA kann, ähnlich wie es für SETUP gefordert wird, durch die enthaltene statistische Expertise Empfehlungen zur Methodenauswahl geben. Jedoch handelt es sich hier lediglich um einen einstufigen Entscheidungsprozeß. Die Datensituation wird nur zu einem Zeitpunkt überprüft und daraufhin eine Empfehlung gegeben. Die Modellierung eines komplexen Inferenzprozesses ist damit nicht erforderlich. Die statistische Expertise in der Beratungskomponente von CORA beschränkt sich also auf einen teilweise wesentlich kleineren Bereich als in anderen Ansätzen, die in der Regel mehrstufige komplexe Entscheidungsprozesse in Form von statistischen Strategien unterstützen.
Eine weitere Parallele zwischen dem System SETUP und CORA besteht darin, daß beide zur Generierung einer Methodenempfehlung datenabhängiges Wissen zugrunde legen. Im Unterschied dazu wird beispielsweise im Rahmen des Systems PANOS (siehe [Wittkowski, 1985]) datenunabhängig eine geeignete Methode ausgewählt. Hier werden statistische Methoden zu Methodenklassen zusammengefaßt und diese dann mit den abgeleiteten Problemtypen in Beziehung gesetzt.
Ein weiteres entscheidendes Kriterium ist die Kommunikation mit dem Benutzer. Eines der ersten statistischen Expertensysteme, das dem Anwender eine grafische Benutzeroberfläche unter Windows bietet, ist das bereits erwähnte System ESTES. Dieses System legt den Schwer-

punkt auf die Dialog– und Erklärungskomponente. Es wird ein sogenanntes Visual Lexikon implementiert, das neben der Möglichkeit zur Erläuterung von Resultaten auch elementare Konzepte des Sachbereichs erklärt (siehe [Steinecker, 1990], Seite 59):

> „Wünscht der Anwender beipielsweise eine Erklärung des Begriffs 'Saisonalität', dann wird neben einer verbalen Definition eine saisonale Zeitreihe erzeugt. Dabei werden sowohl die Werte der Reihe angegeben, als auch eine graphische Darstellung der Zeitreihe produziert. Auf Wunsch können weitere saisonale Reihen angezeigt werden. [...] Darüber hinaus wird es ihm ermöglicht, die erzeugten Zeitreihen zu sichern und als Arbeitsdaten einer eigenen Analyse zuzuführen.“

Auch in CORA kann eine Erläuterung elementarer Begriffe durch das Hilfesystem abgerufen werden. Darüber hinaus wird ein Einblick in die „Analysephilosophie“ ermöglicht. Die gerade zitierten Leistungsmerkmale des visuellen Lexikons von ESTES können aber nicht erfüllt werden, d.h. das Hilfesystem hat hier einen statischen Charakter. Dagegen funktionieren die Beratungs– und Erklärungskomponente von CORA datenabhängig. Zunächst wird mithilfe der Regeln, die von den Dateneigenschaften auf das Verhalten der Schätzer schließen, ein geeigneter Schätzer empfohlen. Die Erklärungskomponente macht diese Auswahlentscheidung des Systems transparent, indem die relevanten Eigenschaften der vom Benutzer eingegebenen Daten angezeigt werden. Eine Erklärung in Form einer Beantwortung von Fragen ist allerdings nicht realisiert. Daher ist ein „Unterrichten“ des Anwenders nur begrenzt komfortabel. Den umgekehrten Aspekt, also das „Unterrichten“ des Systems, betont das oben bereits erwähnte System REX bzw. das darauf aufbauende System STUDENT. Hier bestand das Ziel darin, statistische Strategien durch den Benutzer zu lernen, also neues Wissen aus der Anwendung des Systems zu akquirieren. Dieses Vorhaben konnte jedoch in diesem System nicht vollständig realisiert werden. Ein unmittelbares „Unterrichten“ ist in CORA aufgrund der Trennung von Wissensakquisition und eigentlichem System nicht möglich. Der Wissenserwerb findet in einer externen Wissensbasis mit Unterstützung des Systems MOBAL statt. Lediglich das Ergebnis ist in Form einer Regelmenge in das System CORA integriert. Soll das System mit weiterem Wissen ausgestattet werden bzw. das vorhandene Wissen revidiert oder verbessert werden, beispielsweise durch die Auswertung weiterer Simulationsstudien, so muß die Wissensbasis in MOBAL erweitert bzw. revidiert werden und die durch neue Lernläufe gewonnenen Regeln in die Beratungskomponente von CORA aufgenommen werden (siehe dazu auch Abbildung 6.1).

Neben den oben genannten Kriterien sollen im weiteren einige Aspekte diskutiert werden, die auf den Benutzer des Systems eingehen:

- Welche Zielgruppe wird durch das System angesprochen?

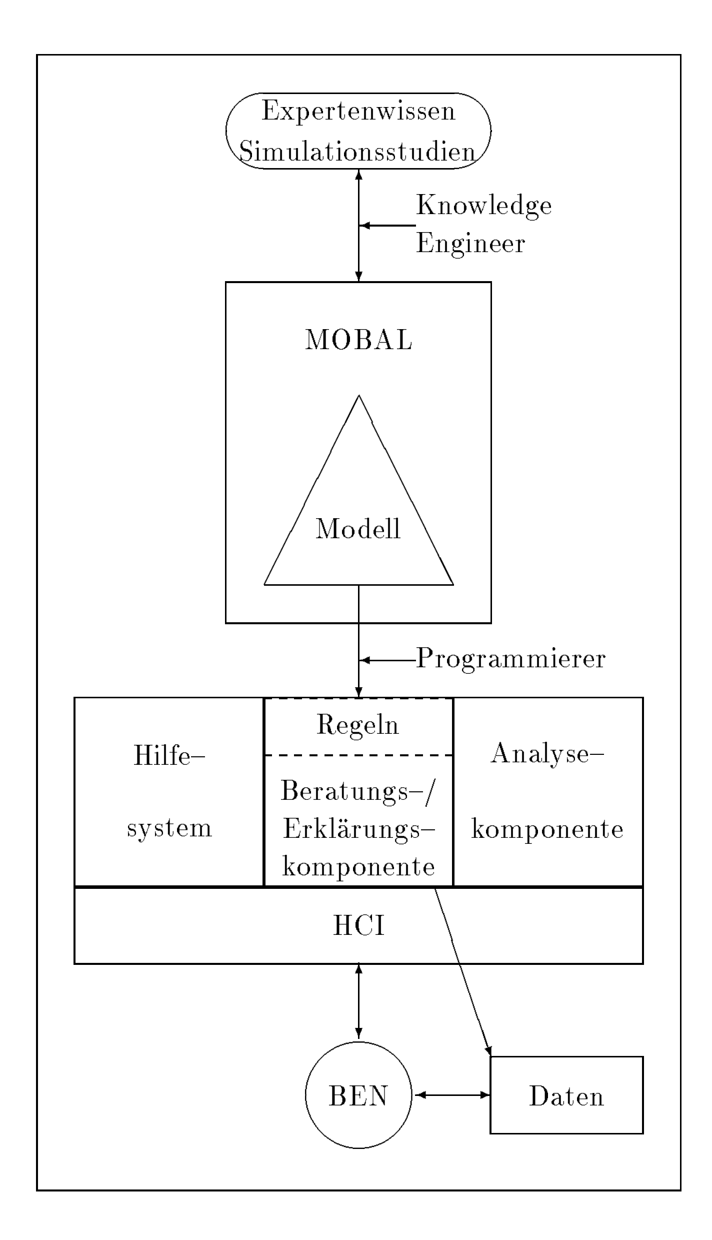


Abbildung 6.1: Wissensakquisition für CORA

- Wird eine Homogenität der Zielgruppe vorausgesetzt oder kann das System unterschiedlich große Vorkenntnisse berücksichtigen?
- Automatisiert das System den Entscheidungsprozeß oder berät es den Anwender lediglich?

Wie bereits mehrfach erwähnt, werden für die Benutzung des Systems CORA nur geringe Statistik–Kenntnisse vorausgesetzt. Da aber für die Durchführung einer Fall–Kontroll–Studie generell die Unterstützung durch eine Statistikerin empfehlenswert ist (siehe 3.1), sollte auch berücksichtigt werden, daß das System von Statistikerinnen verwendet wird. Daraus ergibt sich also eine relativ inhomogene Zielgruppe.
In diesem Zusammenhang wird häufig die Individualisierbarkeit von statistischer Software gefordert (siehe [Steinecker, 1990]). Bei der Entwicklung von CORA wurde darauf geachtet, daß Hilfen für ungeübte Benutzer zur Verfügung stehen, diese aber den Ablauf der Analyse nicht behindern, falls sie nicht benötigt werden. Eng mit diesem Aspekt ist auch der „Autoritätsgrad" des Systems verbunden. Nach Meinung der Autorin sollte ein Benutzer in jedem Fall auch die Möglichkeit haben, alle Entscheidungen letztenendes selber zu treffen, auch wenn das Ziel, eine nicht sinnvolle Anwendung statistischer Methoden zu verhindern, auf diese Weise nicht immer erreicht werden kann. Für restriktive Systeme, die die Entscheidungen selbst treffen, besteht die Gefahr, daß der Benutzer die der Statistik eigene Unsicherheit unterschätzt, die auch durch intelligentere Systeme nicht behoben werden kann. Ein Beispiel für ein relativ autoritäres System ist das bereits erwähnte PANOS (siehe [Wittkowski, 1985], Seite 365):

> „Das Expertensystem [...] wählt durch den Vergleich zwischen externem Problem und implizitem Problemtyp eine Methode aus und ruft diese automatisch auf."

In [Gebhardt, 1988] findet man zu diesem Aspekt den Hinweis:

> „Die allgemeine Gefahr bei DV–Anwendungen, die erst recht für das vielen Benutzern geheimnisvolle Feld der Statistik betrifft, ist die weit verbreitete Computergläubigkeit. Das Expertensystem sollte daher [...] auch die begrenzte eigene Fähigkeit der Problemlösung hinreichend zum Ausdruck bringen."

Diese Forderung wird aber von autoritären Systemen, die den Entscheidungsprozeß automatisieren, in der Regel kaum erfüllt.
In diesem Zusammenhang soll noch einmal der Aspekt des maschinellen Lernens erwähnt werden. Wie insbesondere in Kapitel 4 deutlich wird, ist durch das maschinelle Lernen keine vollständige Automatisierung oder Objektivierung des Wissenserwerbs zu erwarten, was aber fälschlicherweise häufig angenommen wird. Auch hier sind eine ganze Reihe von „Unsicherheiten" festzustellen, die sich auf die Qualität der im System vorhandenen statistischen

Expertise auswirken können. Der Einsatz des maschinellen Lernens wird in diesem Rahmen als Chance zur Verbesserung des Wissens gesehen, die, zumindest in den oben erwähnten Systemen, bislang unberücksichtigt geblieben ist.

Kapitel 7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit ist ein benutzerfreundliches Computersystem entwickelt worden, das Anwender statistischer Methoden speziell bei der Analyse geschichteter 2x2–Kontingenztafeln unterstützt. Diese Unterstützung basiert sowohl auf einem relativ umfangreichen Hilfesystem als auch insbesondere auf einer Beratungskomponente, die einen geeigneten Schätzer für das gemeinsame Odds Ratio empfiehlt. Für die Konstruktion dieser Beratungskomponente wurde auf Methoden aus der Künstlichen Intelligenz zurückgegriffen. Mithilfe des Systems MOBAL wurde eine Wissensbasis modelliert, die eine zur Untersuchung der Eigenschaften von geeigneten Schätzern durchgeführte Simulationsstudie repräsentiert. In dieser Wissensbasis konnte durch maschinelles Lernen eine Reihe von Regeln entdeckt werden, die von den Dateneigenschaften auf das Verhalten der Schätzer schließen. Diese Regeln wurden dann zur Beratung der Benutzer in das System CORA eingebunden.
Die hier gemachten Erfahrungen mit einem solchen Ansatz zeigen, daß der Einsatz von Methoden aus der KI durchaus geeignet ist, detailliertere Ergebnisse aus Simulationsstudien zu gewinnen und in dieser Hinsicht die Qualität der Auswertung zu erhöhen. Nicht nur durch das maschinelle Lernen selbst, sondern gerade durch die Modellierung des Wissens und die Inspektion vorhandener oder eingegebener Regeln konnte die Auswertung umfassend verbessert werden. Auch wenn die in Abschnitt 2.2.5 erwähnten Effizienzprobleme die Arbeit mit dem System MOBAL erschwert haben, so waren die Erfahrungen dennoch insgesamt positiv. Die Begründung liegt insbesondere u.a. darin, daß die Experten die eigene Vorgehensweise bei der Auswertung von Simulationsstudien in diesem Ansatz wiedererkannt haben und die Schritte der Modellierung der Expertise daher nachvollziehen konnten.
Ein weiterer Vorteil des hier verfolgten Ansatzes zur Auswertung von Simulationsstudien ist darin zu sehen, daß nun ohne zusätzlichen Mehraufwand auch wesentlich umfangreichere Simulationsstudien ausgewertet werden können, während für eine Auswertung „per Hand“ die vorliegende Simulationsstudie mit 240 Parameterkonstellationen bereits sehr komplex ist.

Umfassendere Simulationsstudien wären jedoch wichtig, um zu besseren Regeln zu gelangen, die ein breiteres Spektrum an möglichen Datensituationen abdecken.

Desweiteren ist durch die Modellierung der Simulationsstudien deutlich geworden, daß es auch sinnvoll ist, besonders schlechte Eignungen von Schätzern genauer zu untersuchen, um dem Benutzer so „Negativempfehlungen“ im Sinne von Warnungen mitteilen zu können. In diesem Zusammenhang sollte die Expertise durch weitere Regeln ergänzt werden, die sich nicht nur aus der Auswertung von Simulationsstudien ergeben, sondern z.B. auf asymptotischen Eigenschaften der Verfahren beruhen oder sich direkt aus der Berechnung ergeben, also zum Teil datenunabhängige Eigenschaften berücksichtigen. Die hierfür notwendige Ergänzung der Regelmenge ist im Rahmen dieses Ansatzes problemlos möglich.

Ergänzend zu diesen Überlegungen sei im folgenden zudem auf mögliche Erweiterungen der zugrundeliegenden statistischen Expertise hingewiesen. So könnten Zielsetzungen für zukünftige Arbeiten dahingehen, die anderen in der Analyse geschichteter 2x2–Kontingenztafeln vorhandenen Auswahlentscheidungen, wie z.B. die Auswahl eines geeigneten Homogenitäts– oder Unabhängigkeitstests, ebenfalls zu unterstützen. Prinzipiell kann der beschriebene Ansatz auf solche Entscheidungsprozesse übertragen werden. Im Rahmen der vorliegenden Arbeit konnte dieser Aspekt jedoch nicht berücksichtigt werden, so daß sich die Beratung auf die Auswahl eines Schätzers für das gemeinsame Odds Ratio beschränkt. Auch bezüglich der Schichtung ist eine bessere Unterstützung des Benutzers durch weitere Hilfen vorstellbar. Die relativ schwierige Entscheidung, ob davon ausgegangen werden kann, daß die Confounder durch die Schichtung kontrolliert werden konnten, wird zur Zeit weitestgehend dem Benutzer überlassen. Wie in Abschnitt 3.3.1 anhand eines Beispiels gezeigt wurde, können Fehlentscheidungen an dieser Stelle jedoch gravierende Folgen haben, so daß eine Beratung für den Benutzer in dieser Hinsicht ebenfalls sinnvoll erscheint.

Es bleibt aber zu beachten, daß das System so konzipiert ist, daß es erst nach Abschluß des Designs und der Datenerhebung für die Fall–Kontroll–Studie eingesetzt wird. Damit kann der sogenannte Fehler dritter Art ([Kimball, 1957]) nicht ausgeschlossen werden, d.h., daß die Möglichkeit besteht, die „richtige Antwort“ auf die „falsche Frage“ zu erhalten. Das System sollte solche Grenzen der zur Unterstützung der Anwender vorhandenen statistischen Expertise deutlich aufzeigen, indem es beispielsweise auf mögliche Gefahren und Fehlerquellen hinweist. Allerdings war mit der Konzeption dieses Systems zu keinem Zeitpunkt der Anspruch verbunden, die Konsultation eines Statistikers überflüssig zu machen. Eine persönliche Beratung kann und soll ein derartiges Computersystem nicht ersetzen.

Die für ein solches System relativ kurze Entwicklungszeit konnte neben der Beschränkung der im System vorhandenen statistischen Expertise nur durch die Verwendung eines Entwicklungstools wie Delphi erreicht werden. Delphi vereinfacht die Erstellung einer komplexen

grafischen Benutzeroberfläche, aber ermöglicht gleichzeitig auch die Kodierung der Expertise und der statistischen Verfahren. Weitere Features dieses Tools, die zur Konstruktion des Systems CORA noch nicht eingesetzt wurden, gestatten zusätzlich die Einbeziehung grafischer Darstellungen zur Unterstützung der Datenanalyse. Diese sind von Bedeutung, da sie den Bedürfnissen insbesondere der weniger erfahrenen Anwender in hohem Maße entgegen kommen. Daher sollten zukünftige Arbeiten diesem Aspekt, neben den anderen bereits erwähnten, besondere Aufmerksamkeit zukommen lassen.

Literaturverzeichnis

[Borland, 1991] Borland GmbH (Hrsg.): ObjectVision für Windows 2.0, Referenzhandbuch, 1991.

[Borland, 1994] Borland GmbH (Hrsg.): Delphi für Windows 1.0, Benutzerhandbuch, 1994.

[Brachman et al, 1985] Brachman, R.J.; Schmolze, J.G.: An Overview of the KL–ONE Knowledge Representation System. Cognitive Science, Nr. 9, S. 171–216, 1985.

[Breiman et al, 1984] Breiman, L.; Friedman, J.H.; Olshen, A.; Stone Ch.J.: Classification and Regression Trees. Statistics/Probability Series, Wadsworth & Brooks / Cole Advanced Books and Software, Pacific Grove, California, 1984.

[Breslow, Day, 1980] Breslow, N.E.; Day, N.E.: Statistical Methods in Cancer Research. Vol. 1 – The Analysis of Case–Control–Studies, IARC Scientific Publications Nr. 32, Lyon, 1980.

[Buchanan et al, 1984] Buchanan, B.; Shortliffe, E.: Rule Based Expert Systems — the MYCIN Experiments, Addison Wesley, London, 1984.

[Carlsen et al, 1986] Carlsen, F.; Heuch, I.: EXPRESS — An Expert System Utilizing Standard Statistical Packages. Proceedings COMPSTAT 1986, S. 265–270, Physika Verlag, Heidelberg, 1986.

[DIN, 1984] Normenausschuß Informationsverarbeitungssysteme im DIN: Bildschirmarbeitsplätze: Grundsätze der Dialoggestaltung. DIN–Entwurf 66234 Teil 8, Deutsches Institut für Normung, 1984.

[Gart, 1962] Gart, J.J.: On the Combination of Relative Risks. Biometrics, Nr. 18, S. 601–610, 1962.

[Gale et al, 1982] Gale, W.A.; Pregibon, D.: An Expert System for Regression Analysis. Proceedings of the 14th Symposium on the Interface of Computer Science and Statistics, S. 110–117, Springer Verlag, New York, 1982.

[Gale, 1986] Gale, G.: Artificial Intelligence and Statistics. Addison Wesley Reading, London, 1986.

[Gale, 1987] Gale, W.A.: Knowledge–based Knowledge Acquisition for a Statistical Consulting System. International Journal of Man–Machine Studies, Nr. 26, S. 55–64, 1987.

[Gebhardt, 1988] Gebhardt, F.: Statistische Fragestellungen bei einem XPS zur explorativen Datenanalyse. GMD–Studien, Nr. 137, Sankt Augustin, 1988.

[Hauck, 1987] Hauck, W.W.: Estimation of a Common Odds Ratio. In MacNeill, I.B.; Umphrey, G.J.: Biostatistics. S. 125–149, Reidel Publishing Company, 1987.

[Hauck, 1989] Hauck, W.W.: Odds Ratio Inference from Stratified Samples. Communications in Statistics: Theory and Methods, Nr. 18, S. 767–800, 1989.

[Haux, 1986] Haux, R.: Expert Systems in Statistics. Fischer, Stuttgart, 1986.

[Hennekens et al, 1977] Hennekens, C.H.; Rosner, B.; Jesse, M.J.; Drolette, M.E.; Speizer, F.E.: A Retrospective Study of Physical Activity and Coronary Deaths. International Journal of Epidemiology, Nr. 6, S. 243–246, 1977.

[Herczeg, 1986] Herczeg, M.: Eine objektorientierte Architektur für wissensbasierte Benutzerschnittstellen. Dissertation, Universität Stuttgart, 1986.

[Hietala, 1986] Hietala, P.: How to Assist an Inexperienced User in the Preliminary Analysis of Time Series: First Version of the ESTES Expert System. Proceedings COMPSTAT 1986, S. 295–300, Physika–Verlag, Heidelberg, 1986.

[Hoover et al, 1980] Hoover, R.N.; Strasser, P.H.: Artificial Sweeteners and Human Bladder Cancer. Lancet, Nr. 1, S. 837–840, 1980.

[Jida et al, 1986] Jida, J.; Lemaire, J.: Expert Systems and Data Analysis Package Management. Proceedings COMPSTAT 1986, S. 251–258, Physika–Verlag, Heidelberg, 1986.

[Karbach, 1989] Karbach, W.: Entwurfsmethodiken für wissensbasierte Systeme. In Cristaller, T. (Hrsg.): Koordiniertes System von Werkzeugen für die Konstruktion und den Betrieb von Expertensystemen, S. 241–284, GMD–Studien, Sankt Augustin, 1989.

[Kelsey et al, 1978] Kelsey, J.K.; Dwyer, T.; Halford, T.R; Bracken, M.B.: Maternal Smoking and Congenital Malfunctions: An Epidemiological Study. Journal of Epidemiology and Community Health, Nr. 32, S. 102–107, 1978.

[Kietz et al, 1991] Kietz, J.U.; Wrobel, S.: Controlling the Complexity of Learning through Syntactic and Task–Oriented Models. Arbeitspapiere der GMD, Nr. 503, Sankt Augustin, 1991.

[Kimball, 1957] Kimball, W.A.: Errors of the Third Kind in Statistical Consulting. Journal of the American Statistical Association, Nr. 57, S. 133–142, 1957.

[Kleinbaum et al, 1982] Kleinbaum, D.G.; Kupper, L.L.; Morgenstern, H.: Epidemiologic Research: Principles and Quantitative Methods. Lifetime Learning Publications, Belmont, California, 1982.

[Li et al, 1979] Li, S.–H.; Simon, R.M.; Gart, J.J.: Small Sample Properties of the Mantel–Haenszel Test. Biometrika, Nr. 66, S. 181–183, 1979.

[Linos et al, 1980] Linos, A.; Gray, J.E.; Orvis, A.L.; Kyle, R.A.; O'Fallon, W.H.; Kurland, L.T.: Low–Dose Radiation and Leukemia. New England Journal of Medicine, Nr. 302, S. 1101–1105, 1980.

[Mantel, Haenszel, 1959] Mantel, N.; Haenszel, W.: Statistical Aspects of the Analysis of Data from Retrospective Studies of Disease. Journal of the National Cancer Institute, Nr. 22, S. 719–748, 1959.

[Morik, 1989] Morik, K.: Sloppy Modeling. In Morik, K. (Hrsg.): Knowledge Representation and Organization in Machine Learning. S. 107–134, Berlin, Springer Verlag, 1989.

[Morik et al, 1993] Morik, K.; Wrobel, S.; Kietz, J.U.; Emde, W.: Knowledge Acquisition and Machine Learning–Theory. Methods and Applications, Academic Press, London, San Diego, New York, 1993.

[Musen, 1989] Musen, M.: Automated Generation of Model–Based Knowledge–Acqusition Tools. Research Notes in Artificial Intelligence, Pitman, London, 1989.

[Naeve et al, 1987] Naeve, P.; Steinecker, J.: SETUP — Statistisches Expertensystem mit TWAICE und P–STAT. Bericht Nr. 172 der Fakultät für Wirtschaftswissenschaften, Universität Bielefeld, 1987.

[Nelder, 1987] Nelder, J.A.: AI and Generalized Linear Modelling: An Expert System for GLIM. In Phelps, B. (Hrsg.): Interactions in Artificial Intelligence and Statistical Methods. Gower Verlag, Aldershot, 1987.

[Newell, 1982] Newell, A.: The Knowledge Level. Artificial Intelligence, Nr. 18, S. 87–127, 1982.

[Pigeot, 1989] Pigeot, I.: Schätzer des gemeinsamen Odds Ratios in geschichteten Kontingenztafeln. Dissertation, Fachbereich Statistik der Universität Dortmund, 1989.

[Plotkin, 1970] Plotkin, G.D.: A Note on Inductive Generalization. In: Meltzer, B.; Michie, D. (Hrsg.): Machine Intelligence, Nr. 5, S. 153–163, Edinburgh Univ. Press, Edinburgh, 1970.

[Plotkin, 1971] Plotkin, G.D.: A Further Note on Inductive Generalization. In: Meltzer, B.; Michie, D. (Hrsg.): Machine Intelligence, Nr. 6, S. 101–124, Edinburgh Univ. Press, Edinburgh, 1971.

[PG 205, 1993] Projektgruppe 205: Endbericht. Fachbereich Informatik der Universität Dortmund, 1993.

[Quinlan, 1986] Quinlan, J.R.: Induction of Decision Trees. Machine Learning, Nr. 1, S. 81–106, Kluwer Academic Publishers, Boston, 1986.

[Rothmann, 1986] Rothman, K.J.: Modern Epidemiology. Little, Brown and Company, Boston, 1986.

[Schlesselmann, 1982] Schlesselmann, J.J.: Case Control Studies: Design, Conduct, Analysis. Oxford University Press, New York, Oxford, 1982.

[Schreiber et al, 1993] Schreiber, G.; Wielinga, B.; Breuker, J. (Hrsg.): KADS — A principled Approach to Knowledge–Based System Development. Academic Press, London, 1993.

[Shapiro et al, 1979] Shapiro, S.; Slone, D.; Rosenberg, L.; Kaufmann, D.W.; Stolley, P.D.; Miettinen, O.S.: Oral–Contraceptive Use in Relation to Myocardinal Infarction. Lancet, Nr. 1, S. 743–747, 1979.

[Steinecker, 1990] Steinecker, J.: Individualisierbare statistische Software oder warum statistische Expertensysteme nicht das Ende des Weges auf der Suche nach „intelligenterer“ statistischer Software markieren. Volkswirtschaftliche Schriftenreihe, Band 9, Lit Verlag, Münster, 1990.

[Strugholtz, 1993] Strugholtz, H.: Statistische Verfahren zur Homogenitätsanalyse in geschichteten Kontingenztafeln. Dissertation, Fachbereich Statistik der Universität Dortmund, 1993.

[Tarone, 1985] Tarone, R.E.: On Heterogeneity Tests based on Efficient Scores. Biometrika, Nr. 72, S. 91–95, 1985.

[Turner et al, 1975] Turner, K.J.; Baldo, B.A.; Hilton, J.M.N.: IgE Antibodies to Dermatophagoides Pteronyssimus (House–Dust–Mite), Aspergillus Fumigatus and Betalactoglobulin in Sudden Infant Death Syndrome. British Medical Journal, Nr. 1, S. 357–360, 1975.

[Wittkowski, 1985] Wittkowski, K.: Ein Expertensystem zur Datenhaltung und Methodenauswahl für statistische Anwendungen. Dissertation, Fakultät Mathematik und Informatik der Universität Stuttgart, 1985.

[Woolf, 1955] Woolf, B.: On Estimating the Relation between Bloodgroups and Disease. Annals of Human Genetics, Nr. 19, S. 251–253, 1955.